# Task Allocation and Associated Protocol in E-RTA

Shueb Ali Khan[1], Vinit Kumar Sharma[2]

[1]Department of CSE, Shri Venkateshwara University, Gajraula, India.
[2]Department of Mathematics, Shri Ram College, Muzaffarnagar, India.
[1]Department of CSE, KS Vira College of Engineering & Management, Bijnor, India

***Abstract—*** *Agents in E-RTA possess the organizational knowledge that only specifies the list of agents that currently belong to its organization. For joint problem solving agents need to identify team members and do not maintain explicit models of other agents' capabilities. Task allocation protocol's aim is to formulate an automated method for agents to find a 'suitable' agent who is 'available' to perform collaborative task.*

*It is necessary to ensure that the activities of agents always remain coordinated and Joint intentions guide problem-solving activity and play a key role in guaranteeing coordination among agents within an organization in complex and dynamic environments. We also assume that agents possess any time solutions to goals. This is done so that an executing goal can be terminated before its normal completion to avoid priority conflicts of requests.*

*This paper described the task allocation & protocols used for our proposed framework E-RTA. This complete paper described introduction in section 1, task allocation in section 2 in detail. Section 3 deals with temporal conflicts resolution method among intentions. Section 4 describes the results of our experiment and finally conclusion in section 5.*

***Keywords— E-RTA, Joint Problem, Task allocation protocol, problem-solving activity.***

## I. INTRODUCTION

In Multi-Agent-System solving a pool of requests could be in form of different formations i.e. organizations of different agents to balance load and coming rate of requests. In our proposal we aimed to reorganize those formations dynamically during new request or to maintain load.

Problem solving requests arrive at each of these organizations. A request that arrives at an organization is solved cooperatively by agents within that organization and independently of the other organizations. As the rate of arrival of problem solving requests at each of these organizations varies with time, there could be a situation where some organizations may have surplus resources, while others have insufficient resources and thereby turn down problem solving requests. In order to minimize these lost requests, the allocation of resources (agents) to organizations is changed dynamically using the microeconomic approach. This reallocation of resources changes the number of agents in the organizations and their skills, and is intended to balance the demand for resources at each organization with its supply.

But it is necessary to pose the following challenges to the MAS reorganization builder.

- Agents must be able to
  - Adapt to unpredictable changes in problem solving environment, for instance when new information becomes available, it may invalidate existing beliefs or goals and
  - Focus on higher priority tasks.
- The multi-agent system must be able to
  - Adapt to changes in load by diverting resources where they are needed most and
  - Add new agents for problem solving in an incremental manner

and thereby reorganize itself dynamically.

### 1.1 Previous Work

This section explores some of the existing mechanisms for implementing organizational policies. Some of the existing formalisms for implementing organizational policies in multi- agent systems are hierarchical organization, contract nets, social reasoning mechanism, and the use of economic methods.

### 1.2 Hierarchical Organization

In a hierarchical organization, decision-making and control is concentrated in a single problem solver at each level in the hierarchy. Interaction is through vertical communication from superior to subordinate agent. The subordinate agents have no autonomy. It is the superior agents that exercise control over resources and decision- making [30, 29, and 7]. Hierarchical organizations are therefore not suited for autonomous agent interactions.

Kumar et al. in [36] proposed the agent based layered framework for resource allocation. The framework mainly depends on one of the information retrieval technique name called vector space model. The implementation shows by using vector space model implementation as agent using JADE. The two vectors (Query and Total Availability of resources) are given as input and the framework provides the similarity coefficient for the data centers related to specific query.

### 1.3 Contract Net Protocol

Contract net protocol [8,31,9] achieves opportunistic and adaptive task allocation among a collection of problem solvers using a framework called negotiation based on task announcement, bids, and awarded contracts.

Cheng and Ishida [8] investigated the various effects of mutual selection mechanisms on manager and contractor utilities, their analysis (based on queuing theory) produced the following conclusions.

Sandholm [10] describes a variant of the contract net protocol for e-commerce in which tasks can be clustered to allow individual agents to bid for complementary tasks as bundle further extensions were made by Sandholm and Lesser [11,12] where decornmitments and decommitment penalties were introduced.

Other applications of contract net protocol includes Enterprise [13], the transportation control system TRACONET [10] and factory floor operations [14].

### 1.4 Social Reasoning Mechanism (SRM)

This model is based on social power theory [15, 16]. In the SRM model [17,18, 19,20,21,22] dynamic coalitions are formed on the basis of motivation in the form of social dependence relations. In SRM, agents maintain models of acquaintances in a data structure called external description (ED). An ED contains the agent's goals, plans, actions, and resources.

The Mobile Agent is also an option to enhanced adaptability. The paradigm of mobile has two general goals: reduction of network traffic and load balancing. It is majorly being used in enhancement of telecommunication services [32, 33].

Agent cloning [25, 26] is also proposed as a more general approach to agent mobility. In order to obtain a truly adaptive organizational policy we require a complete and comprehensive framework that does the following

- Considers criticality of tasks,
- Allows individual agents to be adaptive, and also
- Allows the multi-agent system to dynamically reorganize itself.

As agents operate in environments where they neither have complete nor correct beliefs about the environment / other agents, it is indeed essential for every agent to have the capability to engage in collaborative problem solving. However, as we are interested in situations with varying problem solving load, it is not enough if individual agents possess team rationality. There should also be a means of having the entire multi - agent system alter its organization in accordance with these variations and continue to provide services as per requirements, always giving preference to higher priority tasks in case of temporal conflicts. A comprehensive framework that satisfies all the above requirements is not available.

## II. TASK ALLOCATION

### 2.1 An overview

At each organization accepts the problem solving requests with priorities and deadlines. The requests arriving at an organization are processed cooperatively by the agents of that organization and independently.

For a joint activity establishment, an agent must firstly recognize the need. 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.

It is a case of individual problem solving, if a goal can be achieved solely by the agent that receives it from the environment. 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. 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-tasks and their temporal orderings, development of a common solution involves finding the actual time at which the sub-tasks can be executed. The fact that agents in a team will obey the responsibility code of conduct is implicit.

Our main target here is to develop a task allocation protocol for OMAS (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. For such conditions, ensuring that each listed conditions is satisfied separately before the commencement of collaborative problem solving, involves a high degree of agent interaction and slow down the speed of operation. To attain this difficulty, the protocol that we propose settles more than one condition in a single message interchange.

Finding a team member and making them agree upon a suitable time is done for every unit-task of the recipe in the temporal order specified by the planner. Priorities are used to resolve any temporal conflicts that arise with already existing commitments, during this process. The lower priority task is either rescheduled to accommodate a more critical task. Deadlines therefore ensure termination of the specified method or the protocol.

Ensuring coordination among team agents, all social activity is represented as a joint intention, which includes the list of team, their roles and the common solution. If an agent decommits a sub-task, 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 jointly dropping a goal if it is not suitable.

The protocol proposed here is time bonded, the planner determines if an anytime solution is available for the sub-t If tasks, it associates a minimum amount of time that needs to be spent for obtaining a meaningful solution. On time elapses, execution of the unit task can be terminated to accommodate more critical requests, else continued to completion.

### 2.2 The protocol

Following assumptions, one, it is assumed that the communication is foolproof and that the message delay time is known to all agents. Two, in order to carry out task allocation activity, agents share a global clock reference. Next, 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 admiration commitments in a convenient manner.

The following notation is used in the discussion that follows

$a_i$ $i^{th}$ action

$A_i$ $i^{th}$ agent

$G_i$ $i^{th}$ goal

$T_i$ the time at which action i is executed

The organizer instantiates a representation of the social goal as a joint intention, in its self-model, as in Figure 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. Figure 1 showing actions, a1, a2, a3, a4, are temporally ordered. The values l1, l2, l3 and l4 indicate the lower bounds on execution time for each of the actions, since actions are assumed to have an anytime solution. This method is however not limited to actions having anytime solutions. If an anytime solution is not available, then li is the fixed period of time that needs to be spent for obtaining the solution. The formulated plan 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, E-RTA associates a default deadline with it. This is done to ensure termination of the task allocation process.

For starting and ending the joint activity, the start time and end time indicate the commonly agreed time. The priority slot indicates the local agent's assessment of the importance of the intention and is used as the basis for computing its disability.

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: $a_1$ $l_1$, $a_2$ $l_2$, $a_3$ $l_3$, $a_4$, $l_4$
Deadline:
Start Time: End Time:
Priority: 23
Status: DEVELOPING SOLUTION / EXECUTING
Contributions:

| | | | | |
|---|---|---|---|---|
| $A_1$ | ORGANIZER | $a_1$ | $t_1$ | AGREED |
| ? | TEAM MEMBER | $a_2$ | $t_2$ | |
| $A_1$ | ORGANIZER | $a_3$ | $t_3$ | AGREED |
| ? | TEAM MEMBER | $a_4$ | $t_4$ | |

**FIGURE 1: View of joint intention**

After identifying 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 fact that some time is required to agree to the solution, work cannot commence immediately and formula for 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 i.e. 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.

Start Time = current time +
2 * number ofnonlocal acts * communication delay -*-
3*number of actions * estimated time to process message
+ communication delay

There is no complete picture, of the capabilities of other agents within the organization, for the team organizer. Hence, 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, and to avoid disordered behavior and many iterations, the organizer takes each action in the recipe in a temporally sequential order. It is to be considering 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 t1=12, t2=16, t3-21, t4=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 a2 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. Then A1 finds a suitable time for a3 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, by broadcasting a task announcement message This message, as in figure 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, that the prospective team member is expected to spend for that action.

Sender: $A_1$
Receiver: all agents within the organization
Type: TASK ANNOUNCEMENf MESSAGE
Contents:
Joint Goal G
Priority 23
Contribution $a_2$
Contribution Time t2
Lower bound on time $l_2$

**FIGURE 2: Task allocation message format**

Upon receipt of proposal the team members evaluate it to see whether it is acceptable; refer to Section 3 for further details of this process If there is no conflict, the agent sets up a joint intention similar to that of Figure 1, and an individual intention as shown in Figure 3 The motivation slot indicates the goal for which contribution is required The status slot is 'pending'.

Action: $a_3$
Motivation: Joint Goal G
Start Time: $t_2$
Duration: Priority
Status Pending Lower bound on time $l_2$

**FIGURE 3: Individual Intentions for Agent A2**

Agents then return a message indicating their acceptance to the team organizer, figure 4. The 'priority of decommited goal' slot indicates whether the prospective team member is able to accommodate the request BV decommiting 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 4: Acceptance bid message**

If 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, figure 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: Modified time bid message**

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 decommiting 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 as figure 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.

After completing an allocated task, the team members report the results of execution to the organizer, as Figure 7.

Sender: $A_1$
Receiver: $A_2$
Type: JOINT SOLUTION AGREED (AWARD)
Contents:
Joint Goal G
Solution

| | | |
|---|---|---|
| $A_1$ | $a^1$ | $t_1$ |
| $A_2$ | $a_2$ | $t_2$ |
| $A_3$ | $a_3$ | $t_3$ |
| $A_4$ | $a_4$ | $t_4$ |

**FIGURE 6: Start of Joint Action (Award Message) notification**

Receiver: $A_1$
Type: RESULT OF EXECUTION
Contents:
Joint Goal G
Priority:
Contribution: a:
Contribution Time: $t_2$
Result:
Time spent executing:

**FIGURE 7: Execution result**

### 2.3 Temporal incompatibilities resolution

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 a1 and a: represented in its self-model (Figure 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.

Before the commencement of their execution a new request arrives that corresponds to the intention as Figure 9.

Name: $a_1$
Motivation: $G_1$
Start Time: 5 Max End Time: 12
Duration: 7 Priority: 10
Status Pending Lower bound on Time 7

Name: a:
Motivation $G_2$
Start Time 12 Max End Time 16
Duration 4 Priority 8
Status Pending Lower bound on Time 3

**FIGURE 8: Consistencies of Intentions**

Name: $a_3$
Motivation: $G_3$
Start Time: 15
Max End Time: 20 Duration: 5
Priority: 10
Status: Pending Lower bound on Time: 5

**FIGURE 9: New Intentions**

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, a2 requires an anytime solution, lower bound < duration, and can therefore be accommodated with the new intention. This requires termination of a2 at time 15 in order to start a3.

In case an anytime solution is not available for a2, then it becomes incompatible with a3. 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, a3 is more desirable Therefore the agent forms the intention to achieve a3 from time 15 to 20 and reschedules a2 after a3. The new intention for a2 now becomes.

**Name:** a2
**Motivation:** G2
**Start Time:** 20
**Max End Time:** 24
**Duration:** 4
**Priority:** 8
**Status:** Pending
Lower bound on Time 4

The other actions of G2 that get affected due to this change also need to be rescheduled. If the new schedule for G2 does not conform to its deadline, then the agent decommits G2 and updates the number of recommitments. This is all that needs to be done if G2 is a primitive goal However if a2 corresponds to a joint goal just rescheduling a2 is not enough. The agent must inform all team members about its decommitment to the originally agreed solution as dictated in figure 10.

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 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 (we will describe resource allocation protocol in our next paper as these papers are taken from our P.hd. thesis work).

Sender: $A_2$
Receiver: All team members
Type: DECOMMITMENT TO SOLUTION
Contents
Joint Goal $G_2$
Priority
Contribution a?
Old Contribution Time: $t_2$ New Contribution Time: $t_2$
Lower bound on time:

**FIGURE 10: Decommitment to solution Message**

Sender: $A_1$
Receiver: All team members
Type DECOMMITMENT TO JOINT GOAL
Contents
Joint Goal $G_2$
Priority
Contribution $a_2$
Contribution Time
Lower bound on time

**FIGURE 11: Decommitment to joint goal message**

## III. EXPERIMENT

The inclusion of anytime solutions results in a considerable improvement in the performance of agents. This can be demonstrated by an experiment as Figure 13. The protocol was implemented in Java and run for two organizations of five agents 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 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.

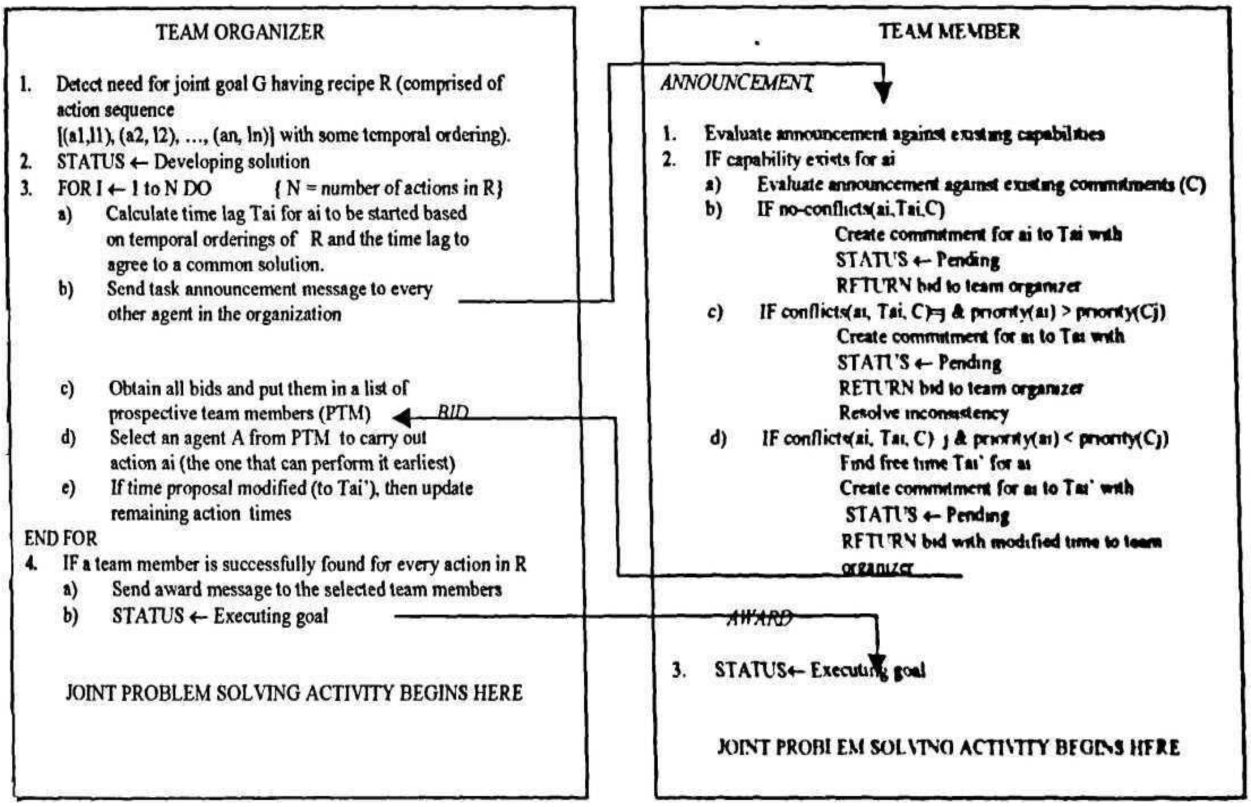


**FIGURE 12: Allocation protocol**

| Task arrival rate n (every *n* sec) | Anytime algorithm % goals decommited | Standard algorithm %goals decommited |
|---|---|---|
| 2 | 46 | 93 |
| 4 | 40 | 55 |
| 8 | 18 | 28 |
| 16 | 0 | 0 |
| 32 | 0 | 0 |

**FIGURE 13: Performance of agents using anytime and standard solutions**

## IV. CONCLUSION

This paper describes the main protocol for task allocation used in our proposed framework E-RTA. All allocation requested tasks have an associated time binding and priority so that higher priority tasks are executed in time by the assigned agents. Even, a situation could arise where an organization is overloaded as the computational load on any organization of the MAS is unpredictable; but the MAS as a whole has the required resources to take on that load.

## REFERENCES


[1] PR Cohen and H J Levesque, "Intention is Choice with Commitment", Artificial Intelligence, 42, pp 213-261, 1990.

[2] PR Cohen and H J Levesque, "Teamwork", SRI Technical Note 504, 1991.

[3] T Dean, M Boddy, "An Analysis of Time Dependent Planning", in Proc. of Seventh National Conf. On AI, pp 49-54 1988.

[4] Reference of our previous paper

[5] Daniel D Corkill, "Hierarchical Planning in a Distributed Environment", In Proceedings of Sixth International Joint Conference on Artificial Intelligence, pp 168-175, Cartridge, Massachusetts, August 1979.

[6] K P Sycara, "Multiagent Systems", AI Magazine, Summer 1998, pp79-92, 1998.

[7] L D Erman and V R Lesser, "A Multi Level Organization for Problem Solving Using Many Diverse Cooperating Sources of Knowledge", Proc. of International Conference on AI, pp 483-490, 1975.

[8] Kumar A., Pilli E. S. and Joshi R. C., "An Efficient Framework for Resource Allocation in Cloud Computing," in 4th IEEE International Conference on Computing, Communication and Network Technologies (ICCCNT'13), Tamilnadu, India, 2013, pp. 1-6.

[9] Cheng Gu, T Ishida, "Analyzing the Social Behavior of Contract Net Protocol", LNAI, 1038 (Berlin: Springer), pp 116-127, 1996.

[10] R G Smith, "Contract Net Protocol High Level Communication and Control in Distributed Problem Solvers", IEEE Transactions on Computers, 29, 12, pp 1104-1113, 1980.

[11] R G Smith, R Davis, "Frameworks for Cooperation in Distributed Problem Solvers", IEEE Transactions on Systems, Man, and Cybernetics , 11,1, pp 61-70, 1981.

[12] T W Sandholm, "An Implementation of the Contract Net Protocol based on Marginal Cost Calculations", AAA1-93, pp 256-262, 1993.

[13] Sandip Sen and Edmund H Durfee, 'The Role of Commitment in Cooperative Negotiation", International Journal of Intelligent and Cooperative Information Systems, 3,1, pp 67-81, 1994.

[14] T W Sandholm, V R Lesser, "Advantages, of Leveled Commitment Contracting Protocol", Tech Report 95-72, Univ. of Mass, Sep 1995.

[15] T W Malone, R E Fikes, K R Grant, M T Howard, "Enterprise: A Market- like Task Scheduler for Distributed Computing Environments", The Ecology of Computation, North Holland, pp 177-205,1988.

[16] H Van Dyke Parunak, "DAI and Manufacturing Control Some Issues and Insights", Decentralized AJ: Proc of First European Workshop on Modeling.

[17] Cristiano Castelfranchi, "Social Power: A Point Missed in Multi-Agent, DAI, and HCl" In Yves Demazeau & Jean Pierre Muller, editors, Decentralized AI, pp 49-62, Elsevier Science Publishers, 1990.

[18] Cristiano Castelfranchi, Maria Micelli, Amedo Cesta, "Dependence Relations Among Autonomous Agents", In Eric Werner & Yves Deamzeau, editors Decentralized AI, 3, pp 215-227, Elsevier Science Publishers, 1992.

[19] Jaime S Sichman, "DEPINT : Dependence-Based Coalition Formation in an Open Multi-Agent Scenario", Journal of Artificial Societies and Social Simulation, 1, 2, March 1998.

[20] J S Rosenchien, "Synchronization of Multi-Agent Plans", Proc 1st National Conference on AI, ppl 15-119, 1982.

[21] Jaime Sichman and Yves Demazeau, "Exploiting Social Reasoning to Deal with Agency Level Inconsistency", In Proceedings of First International Conference on Multi Agent Systems, (ICMAS-95), San Fransisco, USA, June 1995.

[22] Jaime Sichman and Yves Demazeau, "Using Class Hierarchies to Implement Social Reasoning in Multi Agent Systems", In Proceedings of 11th Brazilian Symposium on Artificial Intelligence, (SBIA-94), Fortaleza, Brazil, Outubro, 1994.

[23] Jaime Sichman, R Conte, Y Demazeau, C Castelfranchi ,"A Social easoning Mechanism Based on Dependence Networks", in Proc. of 11 European Conf. On AI, Amsterdam, pp 188-192, 1994.

[24] Jaime Simao Sichman, Yves Demazeau, "A Model for the Decision Phase of Autonomous Belief Revision in Open MAS", Journal of Brazilian Computer Society, 31, pp 40-50, 1996.

[25] Ahmed Karmouch, "Mobile Software Agents for Telecommunications", Guest Editorial, IEEE Communications Magazine, pp 24-25, July 1998.

[26] Marie-Pierre Gervais, "Enhancing Telecommunications Service Engineering with Mobile Agent Technology and Formal Methods", IEEE Communications Magazine, pp 38-43, July 1998.

[27] K S Decker, K Sycara and M Williamson, "Cloning for Intelligent Adaptive Information Agents", in C Zhang and D Lukose (eds), Multi Agent Systems.

[28] O Shehory, K Sycara, P Chalasani, S Jha, "Agent Cloning: An Approach to Agent Mobility and Resource Allocation", IEEE Communications 36, 7, pp 58-67, July 1998.