International Journal of Computing Academic Research (IJCAR)
ISSN 2305-9184, Volume 5, Number 1(February 2016), pp.72-83

http://www.meacse.org/ijcar

A Comprehensive Conflict Resolution Approach for Multiagent Systems

Dimple Juneja[1], Aarti Singh[2], Rashmi Singh[3] and Saurabh Mukherjee[4]
[1]Dronacharya Institute of Management and Technology, Kurukshetra, Haryana, India
[2]Maharishi Markandeshwar University, Mullana-Ambala
[3&4]Banasthali University, Banasthali, Rajasthan, India

Abstract
The paper contributes a novel conflict resolution approach in multiagent systems. The approach proposes methods namely conflict avoidance, conflict prevention and conflict detection, for handling conflicts in any generic interaction protocol in multiagent systems. The proposed approach is comprehensive as it considers avoidance, prevention, detection and resolution of conflict and presents a complete solution.

Keywords: Multiagent Systems, Conflict Avoidance, Conflict Prevention, Conflict Detection, Conflict Resolution, GIPMAS.

1. Introduction

A multiagent System (MAS) [1] is a cooperative effort being offered by many software agents [2]. The agents in MAS have wide variety of capabilities and thus offer variety of solutions to a similar kind of problems. Developments of techniques supporting cooperative design have been left behind irrespective of the developments happening in the domain of MAS. In fact, cooperation among agents in MAS [3] is a tightly coupled endeavor with natural conflicts arising amongst agents while making decisions with the model pertaining to conflict management in agents is lagging [4]. It is obvious that different agents especially agents with heterogeneous capability sets interpret any situation *de- novo*resulting into conflicting decisions about the same task.

In a clustered multiagent system such as GIPMAS [5], different member agents may result into incompatible solution to a common problem. It is also possible that one member agent totally negates the solutions provided by the peer agent. In such situations, conflict occurs and hence the motivation of current work. Moreover, literature depicts very few algorithms [6,7,8,9]offering solutions considering both consensus and conflict simultaneously as these are the two major factors affecting the final outcome. Moreover, conflicting agents have been least addressed and the present conflict resolution solutions make use of weak inferences and thus can't be applied to a generic protocol like GIPMAS.

The current work presents a comprehensive approach for conflict resolution considering GIPMAS as underlying framework. The proposed approach executes in three categories namely conflict avoidance, conflict detection and conflict prevention. The unique contribution is that in case of conflict, it suggests

agreeable solutions proactively and intelligently. Moreover, it offers a second chance to the conflicting agents before taking disciplinary action on the agents.

The paper is structured as follows: Section 2 provides an overview of conflict taxonomy and its need in Generic Interaction Protocol for Multiagent systems (GIPMAS) which serves as backbone of present work.Section 3 presents review of relevant literature in the field of conflict resolution and consensus algorithms. Section 4 elaborates the proposed approach. Section 5 concludes this work.

2. Background

GIPMAS is a clustering based agent interaction protocol in which although agents are grouped on the basis of interests and these exhibit reactive as well proactive cooperative behavior, still these agents get competitive in few situations such as making decisions about selling the product not as per the complete specifications as described by a buyer. Such a situation raises conflict and the same can happen at all levels ranging from global conflicts to local conflicts [10] (see figure 1). A global conflict here implies conflict at domain as well as control levels which further can be narrowed down to conflict between a user agent and executive cluster head (ECH) [5] and conflict between ECH and cluster heads (CH) as well. The local conflict includes conflict between participating and negotiating member agents which may exhibit cooperative as well competitive behavior. On the basis of above, an iterative conflict resolution approach seems to be most suitable as same can be executed at all levels.

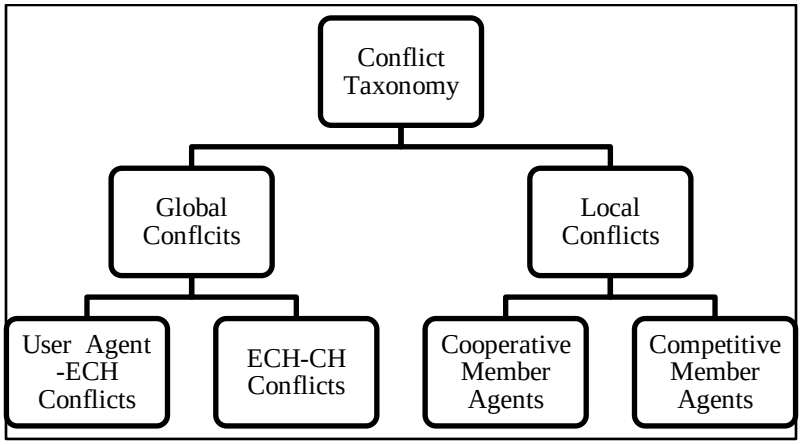

Figure 1: Conflict Taxonomy

While agents in a cooperative design usually remain united as guided by CH, these may get into competitive situations raising conflicts locally. In fact, agents in GIPMAS interactusing RCNTEP [11] and thus inherently acquire reliability value. Higher is the reliability value, higher would be the credibility, thus competition is the result of desire to increase its own reliability value (hence the credibility to remain operational in the system) instead of focusing on credibility of the entire system. Such agents might lose interest in finding the global and optimal solutions unless it adds either to their reliability value or capability set.However, since RCNTEP considers witness trust equally important,if the peer agent becomes self interested (in order to increase its reliability value), its witness trust value would decrease and hence, directly or indirectly, agents are required to work in cooperative mode

inGIPMAS. Hence, this work proposes the conflict resolution strategy focusing on cooperative design inMAS.Next section explores relevant literature in domain of information consensus and conflict resolution formulti-agent systems.

3. Literature Review

Many conflict resolution algorithms [11] such as bluffing , compromise , negotiation , and so on, addressing the conflict resolution problem are available but most of these are application specific i.e. good for a particular MAS for which it has been designed and thus lacks generic applicability. On the other hand member agents may refuse to accept the design of MAS generating domain level conflicts such as acceptance of decisions of CH and ECH. Further, few agents may also be interested in controlling the design directions generating conflict at higher levels (control level conflicts) i.e. between various CH and ECH. It is apparent that conflict at control levels should be avoided and to the best of our knowledge, there seem to be no algorithm supporting this high priority need.

Moraitis and Tsoukias [11] presented a multicriteria approach for distributed planning and conflict resolution for multiagent systems. Agent plans are represented as multicriteria graph and a multicriteria model is used for negotiation purpose. However computing the best path in the individual graph and the negotiation graph is left unaddressed.

Alshabi et al. [12] presented a survey on cooperation coordination, conflict resolution and closely related issues which are critical for the development of large-scale, distributed complex software systems. Their work highlighted the need for a service driven framework for the development of cooperative multi agent systems.

Vasconcelos et al. [13] presented mechanism for the detection and resolution of normative conflicts. Their mechanisms were based on first-order unification and constraint solving techniques which served as the building blocks of moresophisticated algorithms presented by authors for the management of normative positions, i.e. the adoption and removal of permissions, obligations and prohibitions in societies of agents. Their work focused on bothdirect and indirect conflicts between norms. They have presented classic ways for resolving conflicts such as lex superior and lex posterior.

Barber et al. [14] emphasized that equipping an agent with the ability to select their strategies will increase the flexibility and adaptability of system functionality. Their work described strategic decision making, specifically, the representation of strategies and various decision-making approaches.

Pahm and Seow in [15] presented an approach for generating distributed coordination module for multiplediscrete-event agents in the formal languages and automata framework. Their approach presents a formalism called the Distributed Constraint Specification Network (DCSN) that can comprehensibly describe the networking constraint relationships among distributed agents. Their work also presented a plan for multiagent conflict resolution using AND/OR graphs. Tomlin et al. [16] presented a study on conflict resolution for air traffic management. They presented a technique for safe conflict resolution for air traffic using hybrid control system.

Haynes et al. [17] emphasized that groups of agents following fixed behavioral rules can be limited in performance and efficiency. Adaptability and flexibility are key components of intelligent behavior which allow agent groups to improve performance in a given domain using prior problem solving experience. They suggested the usefulness of individual learning by group members in the context of overall group behavior. Their work presented a framework in which individual group members learn cases to improve

the model of other group members.Nwana et al. [18] presented the necessity for co-ordination inagent systems and provided an overview of various co-ordination techniques.Klein and Chrysanthos in [19] presented exception handling approach for agent systems. They highlighted the limitations of "agent-local" approaches for handling exceptions occurring in agent systems, and proposed an approach based on a shared exception handling service. Pappas et al. [20] presented conflict resolution architecture for multiagent hybrid systems focusing on air traffic systems. Their hybrid scheme comprises of a non-cooperative collision avoidance scheme based on game theory along with cooperative scheme based on inter agent coordination.

Sycara in [21] presented a technique to resolve conflicts through negotiations. This work extended the domain of problems toinclude non-cooperative multi-agent interactions whereplanning goals are ill-specified, subgoals cannot beenumerated, and the associated utilities are not preciselyknown. A model for goal conflict resolutionthroughnegotiation had beenimplemented using PERSUADER, aprogram that resolves labor disputes. Giret and Noriega in [22] presented the specification of an agentbasedframework for conflict resolution into OpenMulti-agent Systems by means of grievance protocols.

In order to understand the design requirements of current work, the section critically analyzed the contributions and limitations of existing works. It is discovered that in general following are few of the unfolded challenges which must be countered to find the desired conflict resolution solutions.

1. Agents must be able to evaluate the preconditions and constraints of other agents to avoid conflicts. This further demand that agents should have knowledge of current knowledge bases, updated constraints, configurations and Belief-Desire-Intentions (BDI) of peer agents.
2. Agents should be able to form model of other agents. An approximation of agent's capabilities, reliability values, and authentication certificates, understanding about local and global ontologies is highly desired. In the present scenario, the above approximation is lagging behind.
3. Agents must understand and investigate the original solution, reasons of conflict and should be in the position to generate an alternative solution to counter conflicts and find new agreeable solution.
4. In order to prevent conflicts, agents should be able to negotiate with other agents.

Owing to the above challenges, a comprehensive approach that supports conflict avoidance, detection and prevention at all levels (see figure 2) is strongly desired and is being presented in the upcoming section.

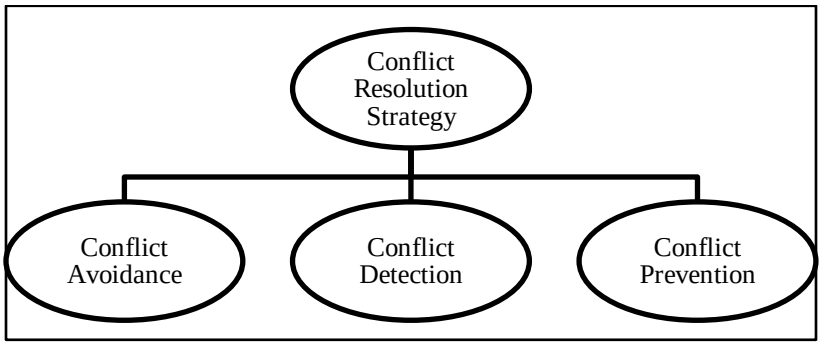

Figure 2: Classification of Proposed Approach

4. The Proposed Conflict Resolution Approach

As is already mentioned that this work considers GIPMAS as the underlying agent interaction protocol. While implementing GIPMAS, it was observed that conflicts usually occurredat all levels when the decisions are fused. Although the consensus protocol [23] also played the role, but at few instances, CH received conflicting results from member agents and hence conflict resolution is desired. However, the proposed approach shall resolve conflicts in all cooperative designs. Three different algorithms pertaining to conflict avoidance, detection and prevention is being proposed. Following are basic design assumptions.

A1. All agents agree to share a set of preconditions before entering into commitment of cooperative problem solving i.e. all agents are possessed with knowledge of preconditions that might raise conflicts.

A2. Agents have access to a knowledge repository that can help avoid conflicts i.e. Registry Agent [5] is accessible to all agents.

A3. If the conflict is detected, agents can get into negotiation directly and settle the conflict on their own also.

A4. However, if agents are not able to negotiate, these relax their constraints and may further negotiate and relax decisions to resolve the conflicts.

A5. If in any case, A1, A2, and A3 fail, decision of conflict resolver agent would be final binding and conflict would then be assumed resolved.

4.1 Conflict Avoidance

The basic idea is here to group the agents in a cluster if and only if all agents eligible to be group members agree to share their preconditions and constraints. Further, when CH distributes the tasks to these member agents, the agents would not raise conflict atleastdue to differences in their initial states. As a result, a conflict safe state would be generated (see figure 3).

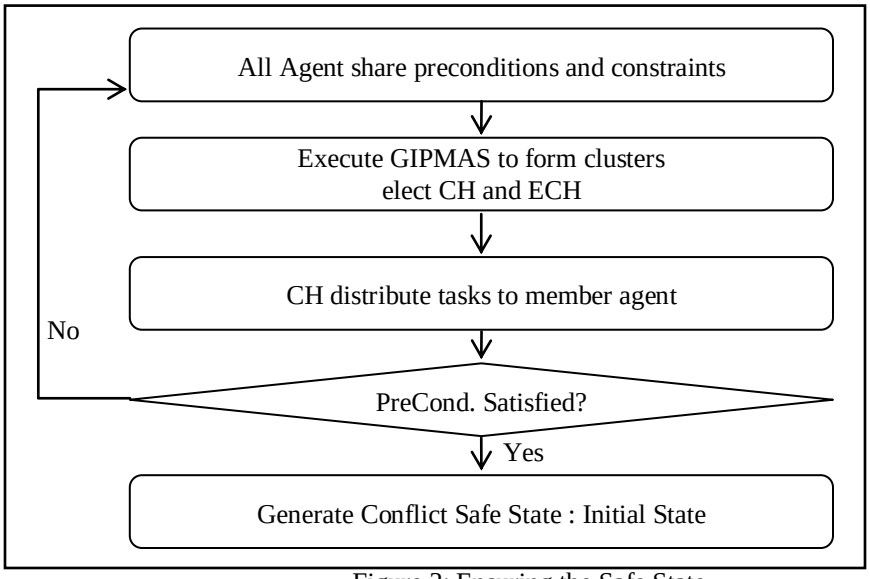

Figure 3: Ensuring the Safe State

Now, assuming that initially the MAS is in conflict safe state, when CH is required to distribute a new task to agents, the agents will not directly and immediately accept the task. Instead, CH only assumes that

task has been distributed and accepted for further processing. The agents then explore the registry agent to evaluate the preconditions of peer agents (involved in the particular task) and also the common shared ontology (see figure 4) and decide if the acceptance and execution of the task would still result into a conflict safe state. If the new task request is safe, the task is accepted for further processing else the agents simply refuse to accept the task. CH then refuse to ECH which further considers forwarding the same to any other CH.

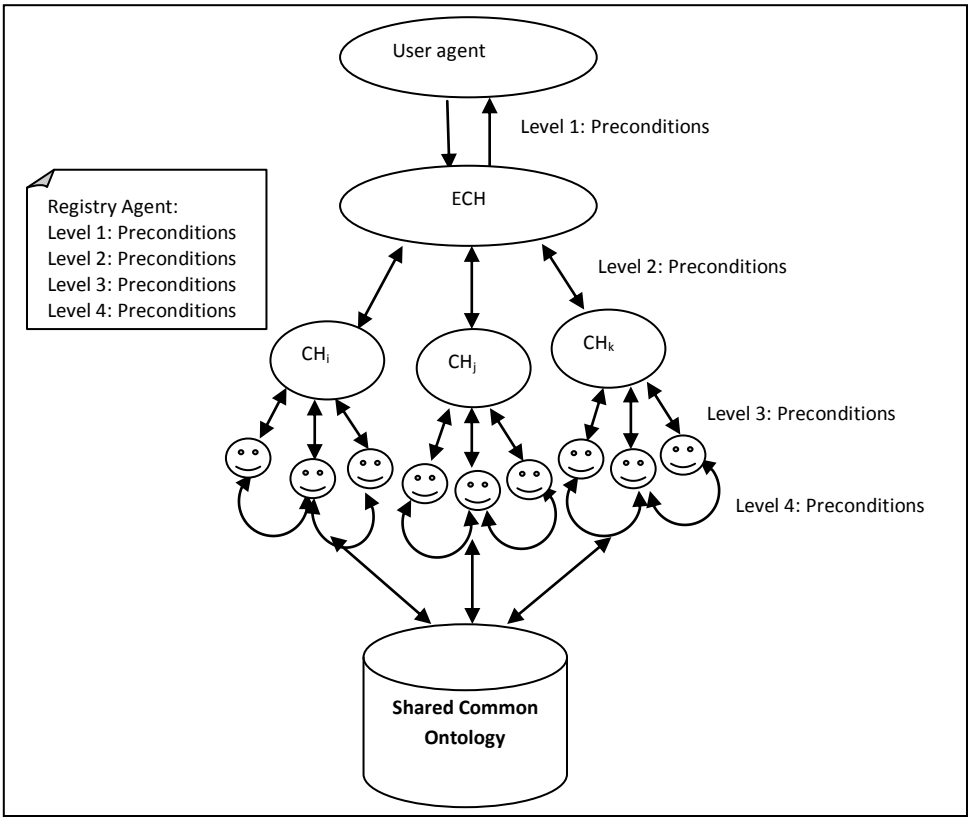

Figure 4: High Level View of Conflict Avoidance Approach

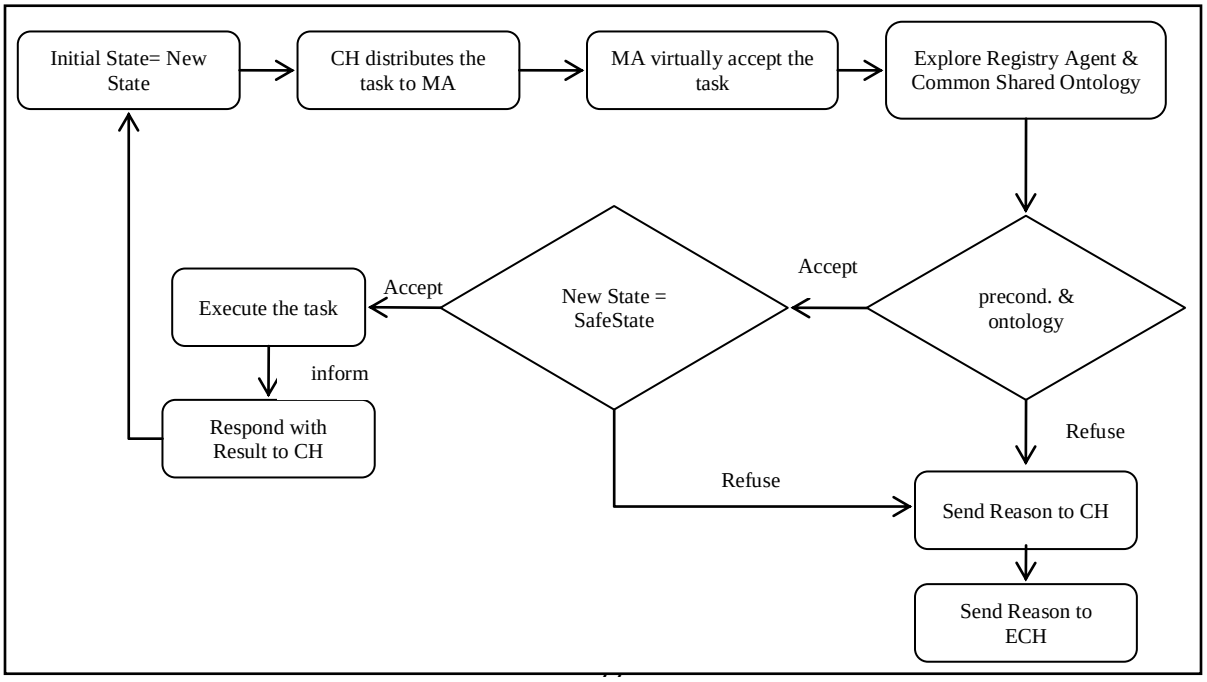

Figure 5: Flow Diagram of Conflict Avoidance Approach

Figure 5 illustrates the working of conflict avoidance algorithm. Now, since the algorithm requires knowledge about all agents, hence it adds to lot of communication and storage requirements thereby conflictthe requirement of light weight agents. Moreover, it merely could avoid the conflict and was unable to prevent the same and hence requirement of conflict prevention algorithm evolved. The conflict prevention algorithm is being presented in the next section.

4.2 ConflictPrevention

In contrast to conflict avoidance approach which allowed distribution of tasks to all agents, the conflict prevention approach allows a task to be first divided into sub-tasks and then distributed among various member agents. Each agent is required to complete their own sub-task and shall submit the result to a common shared memory which is accessible to all agents including CH. The peer agents are then required to vote for the solution thus submitted by the group member. A solution getting more than 50% positive votes shall be accepted by CH else it is discarded and CH may consider redistributing the task among other agents (who had not participated earlier in computing the same task). Since, the agent whose solution was rejected on the basis of voting is not allowed to participate and thus repetitive and conflicting solutions can be easily avoided.If all proposed solutions are acceptable, the CH fuses the results and forwards the same to ECH. Figure 6 illustrates the conflict prevention approach.

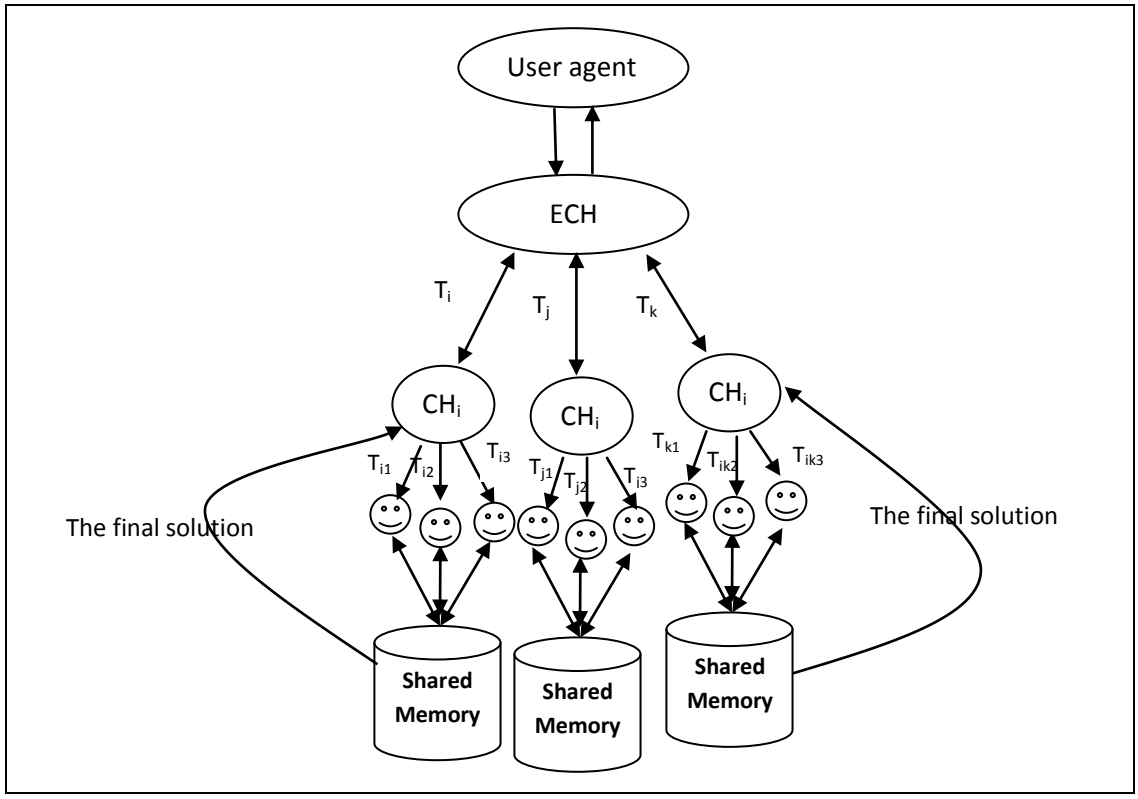

Figure 6: Conflict Prevention Approach

The data structure for saving the results in shared memory is shown in table 1.

Table 1: Data Structure in Conflict Prevention Approach

Agent id	Task id	Proposed Solution	Number of Agents participated in Voting	Favourable Voting Percentage	Status of Solution (Accepted=1, Rejected=0)
Aid	T_{i1}	S	n-1	>=50%	1
Aid	T_{i2}	S	n-1	<50%	0

However, in case, the solution of agent is not acceptable, its reliability value decreases which decreases the overall credibility of the agent. In rarest of case, it may happen that on a continuous decrease in RV value (owing to non-cooperation from peer agents), agent is expelled from the group. Steps for conflict prevention algorithm are listed as follows:

Step1:Divide the task into sub-tasks
Step2:Distribute sub-tasks among member agents.
Step3:Each agent complete their own sub-task and shall submit the result to a common shared memory.
Step4:The peer agents vote for the solution submitted by the group member.
Step5(a): If positive votes>=50% , CH will accept the solution
Step5(b): Else CH discard the solution
CH redistribute the task among other agents (who had not participated earlier in computing the same task).
Step6: All proposed solutions acceptable, CH callsconsensus algorithm to fuse the results and forwards the same to ECH

Both conflict avoidance and prevention technique discussed above are equipped with limitations and are practically inefficient. The third approach i.e. conflict detection approach which initially allows task allocation, execution and interaction of agents and later detect and resolve the conflict seems to be the best approach for handling conflicts in multiagent systems. The same is being discussed in the next section.

4.3 Conflict Detection and Resolution

The approach executes in two phases. In first phase, as shown in figure 7, on detection of conflict at CH level, CH_irejects the result and informs conflicting agents (CA_{ij}) about the status. Here, $i$represents the i^{th} cluster head and j represents the j^{th} member agent of i^{th} cluster. All CA_{ij} are then required to update their own knowledge bases as well as knowledge regarding preconditions of each other. After updating, the conflicting agents interact again and submit the decision to CH. If within three iterations, CH don't declare "no conflict found" and accept the results, the second phase of negotiation between conflicting agents initiates.

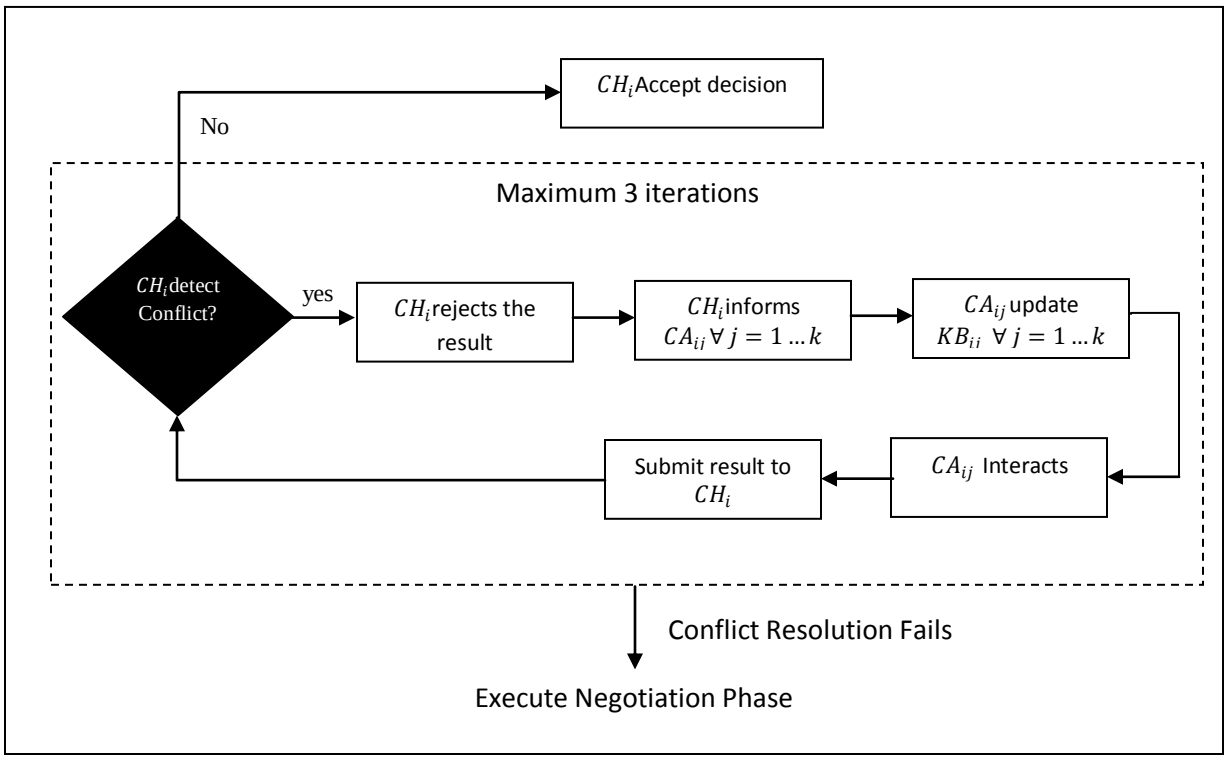

Figure 7: Conflict Detection Phase I

While the existing work supports expulsion of repeatedly conflicting agents from the group, our work uniquely contributes a second chance algorithm for further negotiations illustrated below.

Second Chance Algorithm

Let us assume CA_{i1} and CA_{i2} are two conflicting agents sending information to CH_i. On failure of first phase of conflict detection algorithm, second chance algorithm executes and the interaction between conflicting agents is illustrated in *SecondChanceAlgorithm()* given below in Figure 8.When CH_ideclares "No_Conflict_Found", it calls *CosnensusAlgorithm()*[23] to fuse the results and forward the information to ECH. In the rarest case, it is observed that even after execution of *secondchancealgorithm(),* the agents continue to conflict and in this case only, CH_i would play the role of conflict resolver agent and its decision would be the final binding on all member agents.

SecondChanceAlgorithm()
Input : Conflicting Agents , Conflicting Info
Output: Cooperative Agents, Consensus Info
Conflicting Agents :CA_{i1} ,CA_{i2}
CA_{i1}, CA_{i2}: redefine set of constraints and assign priority
Sort constraints according to priority
While all constraints!=NULL
{
CA_{i1}:Send highest priority constraint to CA_{i2}
CA_{i2}: evaluate the received constraints
If constraint acceptable,
relax its own decision and convey to CA_{i1}.
Else request CA_{i1} to send next high priority constraint
Repeat CA_{i2}
CA_{i1}:If new decision acceptable
convey the final decision to CH_i
Else request CA_{i2} to send high priority constraint
evaluate the received constraint
If constraint acceptable,
relax its own decision and convey to CA_{i2}.
Else request CA_{i2} to send next high priority constraint
Repeat CA_{i1}
}

CH_i: Declares "No_Conflict_Found"
Call CosnensusAlgorithm () to Fuse the results
Forward to ECH
Stop.

Figure 8: Second Chance Algorithm Phase

5.Conclusions

The paper presented a hybrid approach as a solution for issues related to resolving conflicts in MAS. The approach is well suited to avoidance, prevention and detection of conflicts. Majorly, the conflicts arise due to fuzzytasks, indistinctsubgoals andunknown associated constraints. It is further discovered that the issues could be resolved with negotiation and relaxing the constraints imposed by participating agents. Checking of preconditions in conflict avoidance is mutually exclusive which is not very much desired. Further, conflict prevention made the agents more of self-interested behavior instead of being cooperative. Conflict detection appeared to be most suitable approach as the same could overcome the limitations imposed by above two and offered a second chance to agents for negotiating and resolving conflicts. However, it is worth mentioning that all three approaches shall be viewed positively towards cooperative problem solving.

References

[1] Ferber, Jacques. *Multi-agent systems: an introduction to distributed artificial intelligence*. Vol. 1. Reading: Addison-Wesley, 1999.

[2] Genesereth, Michael R., and Steven P. Ketchpel. "Software agents." *Commun. ACM* 37.7 (1994): pp:48-53.

[3] Shehory, Onn, and Sarit Kraus. "Methods for task allocation via agent coalition formation." *Artificial Intelligence* 101.1 (1998): 165-200.
[4] Resmerita, Stefan, and Michael Heymann. "Conflict resolution in multi-agent systems." *IEEE Conference on Decision and Control*. Vol. 3. 2003.
[5] Dimple Juneja,Rashmi Singh, Aarti Singh, Saurabh Mukherjee, "A Clustering-Based Generic Interaction Protocol for Multiagent Systems", Published in proceedings of Third International conference on Information System Design and Intelligent Applications (INDIA-2016), LNCS, Springer (In Press)
[6] Wollkind, Steven, John Valasek, and Thomas R. Ioerger. "Automated conflict resolution for air traffic management using cooperative multiagent negotiation." *AIAA guidance, navigation, and control conference*. 2004.
[7] Bicchi, Antonio, and Lucia Pallottino. "On optimal cooperative conflict resolution for air traffic management systems." *Intelligent Transportation Systems, IEEE Transactions on* 1.4 (2000): 221-231.
[8] Zlotkin, Gilad, and Jeffrey S. Rosenschein. "Negotiation and conflict resolution in non-cooperative domains." *AAAI*. 1990.
[9] Lander, Susan E. "Issues in multiagent design systems." *IEEE expert* 12.2 (1997): 18-26.
[10] Moraïtis, Pavlos, and Alexis Tsoukiàs. "A multi-criteria approach for distributed planning and conflict resolution for multi-agent systems." *Proceedings of ICMAS*. Vol. 96. 1996.
[11] Singh, Aarti, and Dimple Juneja. "An Improved Design of Contract Net Trust Establishment Protocol." *International Journal on Communication* 4.1 (2013): 19.
[12] Alshabi, W., et al. "Coordination, cooperation and conflict resolution in multi-agent systems." *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*. Springer Netherlands, 2007. 495-500.
[13] Vasconcelos, Wamberto W., Martin J. Kollingbaum, and Timothy J. Norman. "Normative conflict resolution in multi-agent systems." *Autonomous Agents and Multi-Agent Systems* 19.2 (2009): 124-152.
[14] Barber, K. S., T. H. Liu, and D. C. Han. "Strategic Decision-Making for Conflict Resolution in Dynamic Organized Multi-Agent Systems." *A Special Issue of CERA Journal* (2000).
[15] Pham, Manh Tung, and KiamTianSeow. "Multiagent Conflict Resolution for a Specification Network of Discrete-Event Coordinating Agents." *arXiv preprint arXiv:1403.4711* (2014).
[16] Tomlin, Claire, George J. Pappas, and Shankar Sastry. "Conflict resolution for air traffic management: A study in multiagent hybrid systems." *Automatic Control, IEEE Transactions on* 43.4 (1998): 509-521.
[17] Haynes, Thomas, Kit Lau, and Sandip Sen. "Learning cases to compliment rules for conflict resolution in multiagent systems." *Adaptation, Coevolution and Learning in Multiagent Systems: Papers from the 1996 AAAI Spring Symposium*. 1996.
[18] Nwana, Hyacinth S., Lyndon C. Lee, and Nicholas R. Jennings. "Coordination in software agent systems." *British Telecom Technical Journal* 14.4 (1996): 79-88.
[19] Klein, Mark, and ChrysanthosDellarocas. "Exception handling in agent systems." *Proceedings of the third annual conference on Autonomous Agents*. ACM, 1999.
[20] Pappas, George J., Claire Tomlin, and Shankar Sastry. "Conflict resolution for multi-agent hybrid systems." *Decision and Control,1996. Proceedings of the 35th IEEE Conference on*. Vol. 2. IEEE, 1996.
[21] Sycara, Katia P. "Resolving Goal Conflicts via Negotiation." *AAAI*. Vol. 88. 1988.
[22] Giret, Adriana, and Pablo Noriega. "On grievance protocols for conflict resolution in open multi-agent systems." *System Sciences (HICSS), 2011 44th Hawaii International Conference on*. IEEE, 2011.

[23] Singh A., Juneja D., Singh R., Mukherjee S., **"**A Clustered Neighborhood Consensus Algorithm for a Generic Agent Interaction Protocol**",** Communicated to International Journal of Advanced Intelligence Paradigms, Inderscience Journals.