COMPLEX NEGOTIATION PROTOCOLS FOR A DISTRIBUTED SIMULATION ENVIRONMENT*

V. RAMACHANDRAN, S. RAMASWAMY, P.K. RAJAN

Software Automation and Intelligence Laboratory, Department of Computer Science and Center for Manufacturing Research

Tennessee Technological University, Cookeville TN 38505. Phone: (931)-372-3691 Email: srini@acm.org / srini@ieee.org

I. Abstract

EMCAP (Electromagnetic Compatibility Analysis Program) is a PC based program used by U.S Navy to assign system frequencies to radars in a single platform or in a group. The central frequency manager (CFM) gets the required operating parameters from all ships and checks whether there is any interference between them considering their relative position. When the probability of interference is high, the CFM assigns to all ships, or to those ships involved in interference, a new frequency. Earlier we reported on an agent-based two-level distributed interactive simulation architecture, designed to detect and resolve interference problems using three distinct approaches; namely master-slave, locally autonomous and negotiation [3], where each ship was simulated as a software agent on a Windows NT machine. The simulation environment was built using JAVA. Under different sets of test conditions, it was observed that the negotiation approach, in general, performed better than the other two modes with respect to the time taken for interference resolution [3-4]. However, a major limitation of this implementation was that the number of ships participating in a negotiation, at any given instant, was limited to two. In a real-world system, conflicts can occur between more than two agents at a time.

II. Introduction

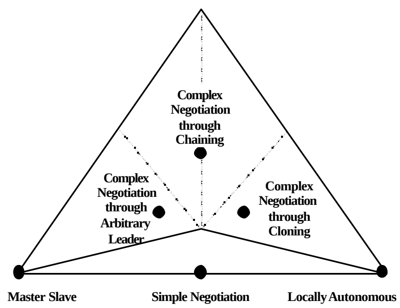

Figure 1. Complex Negotiation Protocols

In this paper, we have developed three complex negotiation techniques, which attempt to integrate the benefits provided by the negotiation approach while incorporating the positive features of the master-slave and locally autonomous approaches for complex negotiations involving multiple participants. The performance of these three implementations is to be evaluated based on the time taken by them to resolve interference. Referring to Figure 1, the implementation in [3-4] included the master-slave, simple negotiation and locally autonomous protocols for coordination. The following three complex negotiation protocols; adapted from other existing literature, have been designed and used in this paper. *(i)* Complex negotiation through arbitrary leadership. *(ii)* Complex negotiation through chaining. (iii) Complex negotiation through cloning. Various factors affecting the complexity of negotiation using these three approaches are summarized in Table 1. In the sequel, these three approaches are explained in detail. In section III, the design of the complex negotiation approaches is explained. All of these approaches are an enhancement to the existing two party negotiation approach [3], designed to involve additional members in a negotiating group. In section IV, the implementation details of these approaches are discussed. Section V concludes the paper.

Types of Complex Negotiation (→) / *Factors Affecting Complexity of Negotiation (↓)*	*Arbitrary Leader Approach*	*Chaining Approach*	*Cloning Appraoch*
Increase in the number of agents in the group	Direct impact	Not a direct impact	Not a direct impact
Number of Agents in the negotiation	Not a direct impact	Not a direct impact	Direct impact
Order of entry into the group	Not a direct impact	Direct impact	Not a direct impact

Table 1. Factors Affecting the Complex Negotiation Approaches

* This work was carried out at the Software Automation and Intelligence Laboratory in the Department of Computer Science at Tennessee Technological University as part of our research on intelligent coordinating entities, or ICE. S. Krishnamurthy is a graduate student in the Electrical and Computer Engineering Department at Tennessee Technological University. Dr. S. Ramaswamy is an Associate Professor and Chair of the Computer Science Department at Tennessee Technological University. Dr. P.K. Rajan is the Chair of the Electrical and Computer Engineering Department at Tennessee Technological University. Phone: (931)-372-3691. Email: srini@acm.org / srini@ieee.org..

III. Design of Complex Negotiation Approaches

III.A. Negotiation through an Arbitrary Leader

In this method of complex negotiation, an arbitrary leader is selected for arbitrating the interference resolution process. This method of negotiation is similar to negotiation among agents through a persuasion process [5]. When there are ‘n’ ships, then ‘n-1’ negotiations have to take place before a leader is chosen to resolve the conflict. Once it is determined that the number of ships in the conflicting group is greater than two, then the complexity of this negotiation process would remain the same, irrespective of whether 3 ships are involved in conflict or all the ships in the group are involved in conflict. Hence, as the number of ships in the group increases, the complexity of this negotiation protocol increases. This approach is illustrated in Figure 2, which shows the hierarchical picture of the complex negotiation process between 8 ships, i.e. s1, s2, s3, s4, s5, s6, s7 and s8. Anyone of the 8 ships can become the arbitrary leader for the group and this can be decided in 7 simple handshakes between these ships. For example, if interference occurs between s2, s4, and s7 then s(1,8) assigns new interference free frequencies to these ships. The first step in this negotiation strategy is to determine the number of ships in the conflicting group. If this number is greater than two, then all the even numbered ships (e.g. s2, s4, s6, etc.) in the group undergo simple negotiation with their immediate predecessor ship and selects a master for themselves; namely s(1,2),s(3,4),s(5,6) and s(7,8). Here there are 8 ships initially and after the initial handshakes, there would be four "possible" master ships for the group. Now s(1,2) and s(3,4) select a master among themselves; say s(1,4). Similarly s(5,8) will act as the master for s(5,6) and s(7,8). In the final step of this negotiation process, s(1,4) and s(5,8) select a master among them; namely s(1,8), that has the results of all simple negotiations that happened earlier in selecting s(1,8) as the arbitrary leader for the group. Ship s(1,8), which acts as the master to all the agents can be any one of the 8 ships and hence is called a negotiation through an arbitrary leader.

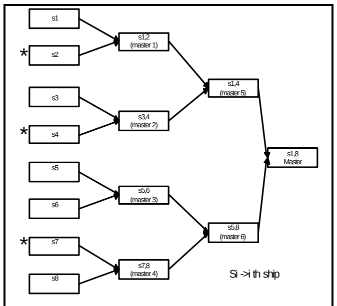

Figure 2. Negotiation through an Arbitrary Leader

This election process can be repeated either on demand or periodically, to elect a different master at different periods of time. The advantage is that specific real-time conditions could influence local decisions; hence paving the way for a scenario based election process.

III.B. Negotiation through Chaining:

In this method, a rank is assigned to each of the ships based on when they join the group. Every ship reports its radar frequencies and its time stamp to the previous ship (time stamp of a ship refers to the time at which the ship joined the group). Hence, each ship will have the frequencies of all other ships that joined later. During interference, the ship with a higher rank among the interfering ships resolves the interference and ensures no further interference by negotiating with the highest ranked ship before passing on the new interference free frequency. First, a master among the interfering ships is decided and depending on the time stamp of this master, the number of negotiations required to resolve the interference is determined. The number of ships in the group does not affect the complexity of this method, but they depend on the time stamps of the ships involved in negotiation. This approach is illustrated in Figure3, which depicts in detail the ships having information about other ships. When interference occurs between 3 ships, then the ship that has joined the group at the earliest among the three ships becomes the master coordinator for them and resolves the interference. For example, suppose s3, s5, s8 have interference, then after comparing the time stamps of these three ships, it is determined that s3, which joined the group earlier than the other two ships, becomes the master. Ship s3 allocates new frequencies to s5, s8. Before sending these frequencies to s5, s8 and s3 negotiates with s1 by passing these newly decided frequencies to s1, in order to make sure that the new frequencies decided by s3 does not interfere with any other ship.

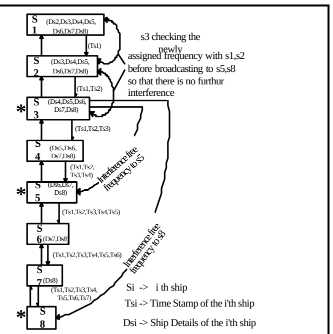

Figure 3. Negotiation through Chaining

This approach provides for a strict hierarchical structure based on the time of entry into the group. Ships can be promoted and demoted in a group by a ship leaving and rejoining the group at a later time.

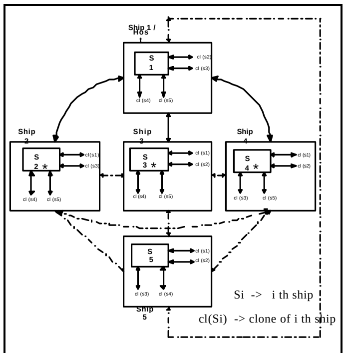

Figure 4. Negotiation through Cloning

III.C. Negotiation through Cloning

In this approach, each ship creates a clone and passes them to every other ship in the group.

When the need for negotiation arises, each ship negotiates with the clones of all other ships. The multiple results of these negotiations are then compared at one location and a consensus for the group is reached [6]. All the clones of a particular agent are updated at the same instant of time. Hence, it is assumed that data on all clones of an agent are synchronized. In this method, the complexity of negotiation will depend on the number of ships in the conflicting group. The total number of ships in the group or the time stamp of each ship in the group will not affect the complexity of negotiation. Figure 4 shows a system having 5 agents implementing this approach. The first ship in the group is considered as a host. A clone of each ship is passed on to all

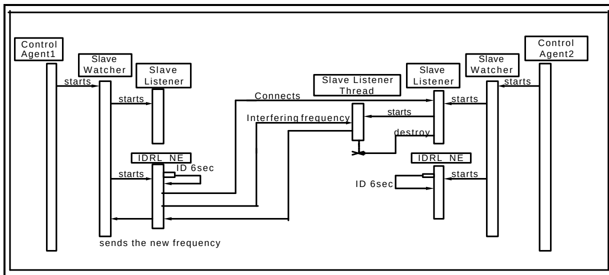

Figure 5. Simple Negotiation Protocol

incoming ships. When interference occurs between radars on ships s2, s3 and s4, each of these ships would resolve themselves to an interference free frequency after negotiating with the clones of other ships. Now s2, s3, s4 will report to host, which checks for any interference between s2, s3 and s4. If interference is detected, these ships again resolve to a new frequency. Once the host confirms that there is no interference between s2, s3 and s4, then the interference-free frequencies of s2, s3 and s4 are updated to their respective clones.

With this approach, an attempt is made to reduce the instant network dependencies caused by a interference. The synchronization of the databases could be a periodic process that does not affect the complexity in a steady-state situation.

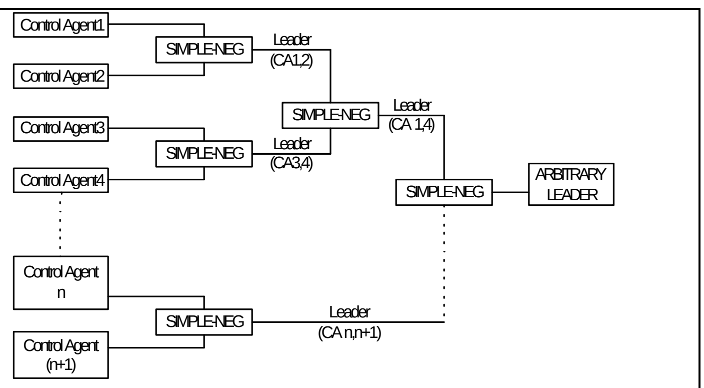

Figure 6. Interaction Diagram for Negotiation through an Arbitrary Leader

IV. Implementation of Complex Negotiation Approaches

This section explains the implementation details of the complex negotiation approaches. Figure 5 presents the simple negotiation protocol that is currently used in the simulation environment. During the simple negotiation process, the control agent starts the SlaveWatcher class, which in turn starts the SlaveListener and the IDRL_NE classes. The SlaveListener class runs in an infinite loop ready to accept client connections. The IDRL_NE class has the interference detection and resolution algorithms in it. In the sequel we will present the current status of the implementation of the above three approaches. This thread again runs in an infinite loop and checks for interference every six seconds. If control agent 1 detects that it is having interference from control agent 2 then it connects to the SlaveListener of control agent 2 and starts negotiation. The SlaveListener starts a new thread SlaveListenerThread in order to negotiate with control agent 1. The SlaveListener thread then randomly generates a new interference free frequency and sends it to control agent 1[4].

IV.A. Complex Negotiation through an Arbitrary Leader

Figure 6 shows the interaction diagram for complex negotiation mode through an arbitrary leader As seen in this figure the control agents undergo simple negotiation among themselves to elect an arbitrary leader. In this implementation as the number of control agents increase the complexity of this approach also increases but the complexity remains unaffected with increase in the number of control agents in the conflicting group. The impact of various factors affecting this approach is shown in Table 1.

IV.B. Complex Negotiation through Chaining

In this approach, a master among the interfering control agents is chosen first depending on the rank of the conflicting control agents. In the interaction diagram of this approach shown in Figure 7, Control agent 1 refers to the master among the interfering agents. Here we assume that three agents are involved in the conflict. Each control agent starts the SlaveWatcher class, which in turn starts the SlaveListener and the IDRL_CH classes. The SlaveListener class runs in an infinite loop ready to accept client connections. The IDRL_CH class has the interference detection and resolution algorithms in it. This thread again runs in an infinite loop and checks for interference every six seconds. During an interference resolution process, Control agent1 that acts as the master to the other two control agents resolves a new frequency for them. Before sending these new frequencies to control agent 2 and 3,the master undergoes simple negotiation with all those control agents that have joined the group earlier than itself. Once Control agent1 ensures that there is no further interference they broadcast the new frequency to other member in the conflicting group. As mentioned in Table 1, the complexity of this approach depends only on the order of entry of control agents in to the group. If the master among the conflicting members has no control agent ranked higher than itself then this method of implementation could be the fastest as the complexity would remain minimum during this condition.

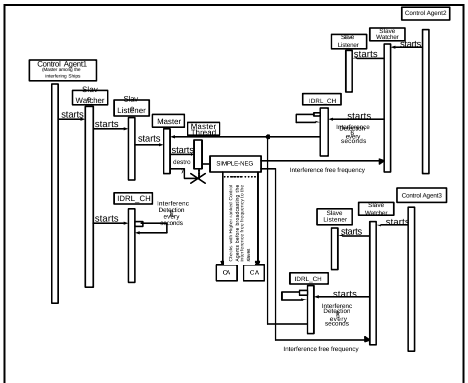

Figure 7. Interaction Diagram for Negotiation through Chaining

IV.C. Complex Negotiation through Cloning

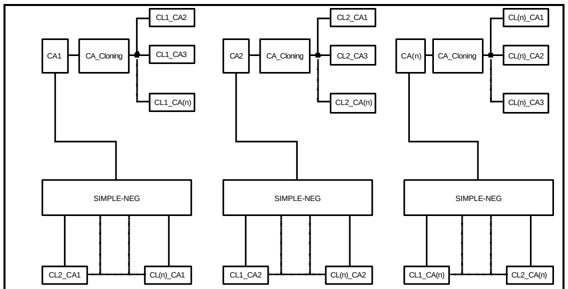

Figure 8. Interaction Diagram for Negotiation through Cloning

The interaction diagram of complex negotiation through cloning is shown in Figure 8. CL1_CA2 refers to the clone of control agent 1 that will be sent to control agent2. The same convention is maintained with all other clones. In the interaction diagram shown it is assumed there are ‘n’ control agents and CA (n) corresponds to the n^{th} Control Agent. The first control agent is assumed to be the host and it is the place where checking of interference among the newly resolved frequencies is done. Each control agent in the group will start the CA_Cloning class to produce the clones. During interference, the conflicting agents negotiate with the clones of every other agent to resolve themselves to a new interference free frequency. When the conflicting agents obtain a new frequency they check with the host in order to ensure no further interference between these new frequencies. In this implementation as the number of control agents in the conflicting group increase, the complexity of this approach is also expected to increase. As mentioned in Table 1, the complexity of this approach is not affected by either the number of agents in the conflicting group or by the total number of control agents in the group.

V. Conclusions

In this paper we have presented three complex negotiation approaches for resolving interference problems among mobile radar units. A comparative evaluation of these three approaches with the results presented in [4] will be performed to determine the effectiveness of the proposed complex negotiation protocols.

VI. References

1. Habib Adnan, S. Ramaswamy and R. Macfadzean, " An Agent-Based Distributed Simulation Environment for Frequency Assignment in Mobile Radar Units", *1998 Summer Simulation Conference*, Reno. NV, July 1998.
2. K. Srinivasan, J. Cherry, T. Scalf, N. D. Cannon, S. Ramaswamy, “A Two-level Distributed Interactive Simulation Architecture for Radar Frequency Assignment” *1999 Summer Computer Simulation Conference*, Chicago, IL, July 1999.
3. K. Srinivasan and S. Ramaswamy, " Implementation and Evaluation of a Distributed Interactive Simulation Architecture for Group Interaction and Coordination: A Case Study in Interference Detection and Resolution in Naval Radar Units", *2000 Summer Computer Simulation Conference*, Vancouver, Canada, July 2000.
4. S. Ramaswamy, K. Srinivasan, P. K. Rajan, R. MacFadzean and S. Krishnamurthy, " A Distributed Agent-based Simulation Environment for Interference Detection and Resolution", *Special Issue on Software Agents in Simulation, SIMULATION. June 2001, to appear*.
5. Takayuki Ito and Toramatsu Shintani, "Persuasion among agents: An approach to implementing a group decision support system based on Multi-agent Negotiation", Department of Intelligence and Computer Science, Nagoya Institute of Technology. http://citeseer.nj.nec.com/update/162201, retrieved on Feb 28,2000.
6. Katia Sycara, Takayuki Ito and Toramatsu Shintani, "Multiple negotiations among Agents for a Distributed Meeting scheduler", *Proceedings of fourth international conference on Multi-agent systems (ICMAS’ 2000)*, http://www.cs.cmu.edu/~softagents/papers/tora-icmas00-poster.pdf , retrieved on Feb 28,2000.