Implementing Heterogeneous Agents in Dynamic Environments, a Case Study in RoboCupRescue

Jafar Habibi, Mazda Ahmadi, Ali Nouri, Mayssam Sayyadian, and Mayssam M. Nevisi

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
habibi@sharif.edu
(m_ahmadi,nouri,sayadian,mayssam)@ce.sharif.edu

Abstract. Design and construction of multi-agent systems is a challenging but an intriguing problem. It is because of the intrinsic distribution of the intelligent components. In such environments the interaction and communication between the constituent parts extends the complexity since appropriate coordination methods need to be designated and employed. In this paper a successful experiment in designing and implementing such an environment is presented [1]. The test bed for this research is the rescue simulation environment. The architecture of the implemented heterogeneous agents takes advantage of various algorithms. These algorithms make the agents act intelligently by themselves albeit they happen to act quite in coordination with each other. The implemented algorithms for the sake of cooperation between the heterogeneous agents enhance the overall pay off of the system. The autonomy of the agents is guaranteed by means of some methods such as reinforcement learning, decision trees and some sort of heuristic functions. In order to settle the agents in coordination with each other and make them act cooperatively, some other methods have been applied. Among these methods, combinatorial auctions, coalition formation, function approximation for evaluating the value of cooperation, and some probabilistic and heuristic methods can be named.

1 Introduction

Nowadays design and construction of multi-agent system infrastructures is a challenging but an interesting problem. Designing systems for soccer player robots[5], computer-aided design of a generic robot controller for a multi- robot system[9], design and implementation of automated highway systems[2], and the hot topic of trading agents[8] are a few examples of the works in this field. In this paper we will present our experiment in the design and implementation of a complex environment of multi heterogeneous agents, so that it can be used for further similar activities.

[1] The implemented agents won the first place award in RoboCup Rescue Simulation league in Fukuoka, Japan, 2002.

M. Schillo et al. (Eds.): MATES 2003, LNAI 2831, pp. 95–104, 2003.
© Springer-Verlag Berlin Heidelberg 2003

The engaged test bed is the rescue simulation environment. This test bed is basically designed for the goal of disaster mitigation of an earthquake. Three kinds of completely different agents are aimed to minimize the overall damage to the city. Such agents have various abilities and hence different responsibilities such as extinguishing burning buildings, rescuing injured civilians, etc. Also they are supposed to come across a mutual agreement so that their cooperation and coordination would enhance their collaborative efforts and this adds to their complexity. Two aspects of a multi-agent system with intelligence are eligible to note. The first one is the intelligence of each agent - the micro level issue. The other one is considered with the system as a whole and it is the agents' coordination and cooperation to reach desired goals. In the implemented system both issues are considered and emphasized. It means that although each agent tries to perform his assigned tasks as perfect as possible, he tries to act so that the overall system benefits. In other words the agents are not selfish.

This paper is organized as follows: After introduction we present some general explanation of the rescue simulation environment as our test bed. In section three the fire brigade algorithms are presented. In section four the topics related to police agents are discussed. In section five ambulance team algorithms are presented. Inter-agent communication methods and related subjects are discussed in section six. In section seven a comparison of experimental results is presented and we come to conclusion in that section.

2 Overview of RobocupRescue Simulation

RoboCup Rescue Simulation is one of the competitions in RoboCup. The main aim of RoboCup Rescue Simulation is simulating a disaster situation in a city. There is a kernel simulating the city and some simulators simulating the disaster conditions[4,6]. The parts that we have developed are the agents. The agents are:

- Fire Brigades
- Ambulance Teams
- Police Forces
- Head Quarters

The main goal of the agents is to rescue more civilians[4]. Although ambulances are responsible for rescuing civilians, but polices will clear the roads so that ambulances and fire brigades can move in the city. Fire brigades have to extinguish fires to reduce the amount of damage (the less fire, the more alive civilians).

3 Fire Brigade Agents

In this environment, the fire brigades are responsible for controlling the spread of fire in the city, and extinguishing as many buildings as possible. For this purpose, each agent takes advantage of his visual perception and identifies the buildings on fire. For each burning building the agent autonomously tries to estimate how

dangerous that building would be and how much it threatens its neighbors. After this phase, the fire brigades need to act upon the world's situation in a unified approach to increase their coordination. The most obvious approach in this phase is finding the most important buildings on fire and extinguishing them. So, the way agents calculate a building's priority plays an important role in this phase.

A proposed workflow for a fire brigade is depicted in figure 1. As the figure suggests, the workflow contains four phases, namely: perception, analysis, decision making, and implementation. This means that the agent first receives raw information about the environment. Then by means of communication with other agents, his experiments and his experiences, the agent uses this information to gain some kind of knowledge that would be useful in the decision making phase. Then the agent's world model is investigated to find appropriate targets. The most useful and the best estimated targets are selected in this decision making process. In the last phase the agent implements the desired actions according to the target he has chosen.

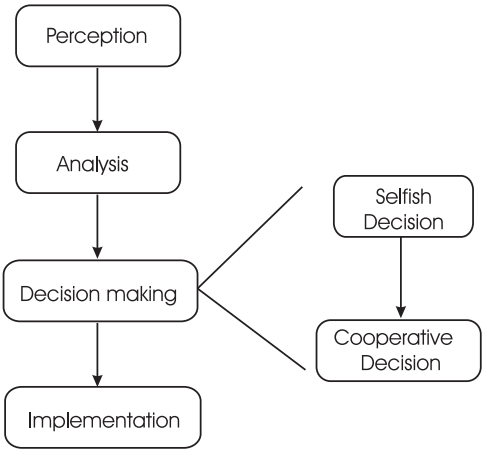


Fig. 1. Agent's workflow

The decision making section is the most important phase in the agent's workflow in each cycle. The fire brigade agents use a two layered architecture for this phase. In the first stage, the agents decide independently. They do not care the state of other agents and they selfishly choose some targets for their own. In the next stage, the agents try to both actively coordinate with other fire brigades and also communicate with other types of agents. In this way the overall rescue integrity is guaranteed and this collaboration enriches the result and the overall performance of the system. One of the advantages of this architecture is the independence of the two layers. This enables implementation and evaluation of different algorithm in each layer.

3.1 Preliminary Decision Making Based on Agents' Selfishness

The implemented agents estimate the buildings' priorities in an individual manner and without considering the existence of other agents. The buildings are sorted afterward based on their priorities and a few of the most critical ones are selected from the many available. The values according which the importance of buildings are calculated, mainly includes two types of criteria; The intrinsic value of a building, which is a measure of importance in the city independent of the situation of the disaster (e.g. geographical location of the building in the city) and the strategic importance which, on the other hand is dependant on the relative location of the buildings according to the distributed sites of burning buildings[2].

This list of candidate buildings is then fed to the next layer of decision making where the targets are assigned to the agents so that the cooperation and coordination between them is also guaranteed.

3.2 Improvement in Decision Making, Seeking Coordination, and Cooperation

While designing fire brigades, it is very important to keep in mind the significance of the coordination between them. Since buildings are extinguished much faster if multiple agents are assigned to them, our agents tend to form coalitions with each other in order to achieve a higher performance.

As there are too many targets and too many agent combinations, the number of possible coalition structures increases drastically and therefore brute force checking of all the combinations is not feasible. In fact, achieving an optimal solution with the help of approximation is inevitable. For this purpose, the agents are regarded as resources that will be assigned to burning buildings. This approach leads us to view the problem from a combinatorial auction perspective.

The combinatorial auction problem has been emphasized in research recently mostly in combinatorics[3]. In this problem agents are allowed to bid directly for bundles of resources. Suggesting a set of resources, the agents propose a value for them. Then the auctioneer is assumed to allocate resources to bidders so that the overall benefit of the system is maximized.

In the proposed system, the auctioneer is simulated so that the optimal resource allocation and coalition formation is achieved. In this system the buildings are assumed to play the role of bidders and agents are regarded as the extinguishing resources. So it can be assumed that buildings suggest values for a bundle of fire brigades. To evaluate each bid the followings are important:

- The importance of the building which is an indicator of its strategic and intrinsic importance.
- The amount of time it takes the proposed coalition to put out the building. This measure can be calculated by means of statistical and computational analysis.

[2] A group of connected burning buildings is called a site of fire

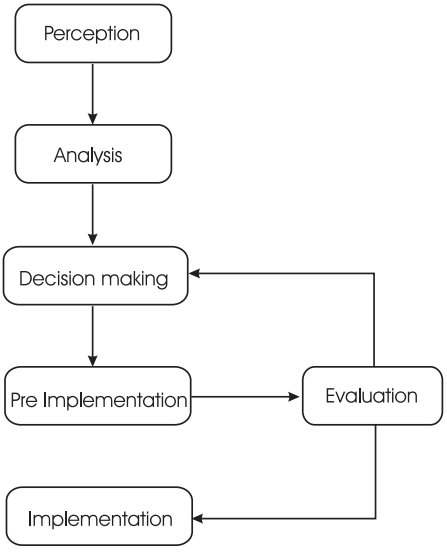


Fig. 2. Agent's workflow with feedback

The most complex part of the combinatorial auction is the winner determination algorithm for the auctioneer. As a matter of fact, the most difficult part of the combinatorial auction simulation is to find an optimal coalition structure among the many possible, so that the overall payoff of the system is maximized. Since the algorithms for this problem are NP, in this system the approximate algorithms developed by Edo Zurel have been exploited. This algorithm uses some heuristic functions to prune the search space very fast. Then it uses some greedy methods and by means of hill climbing it finds a very good coalition structure which is an approximate of the optimal solution[11]. This way by simulating a coalition formation via combinatorial auction for the fire brigades the inter-agent coordination is guaranteed in a satisfactory manner and it is very close to the optimal solution.

3.3 More Improvement in Decision Making

The mentioned agent workflow in figure 1 suffers a major drawback: In decision making phase the fire brigades try to select the best possible coalition structures and as to collaboratively extinguish the target buildings. As a matter of fact these coalition structures are formed in the implementation phase. The performance of the system is tightly related to the successful implementation of these coalition formations. According to this workflow, no feedback is given to the decision making phase so it is not too surprising if a selected coalition structure can not be formed in the next phase. This can be as a result of some problem such as path planning. Hence all the agents that are supposed to form that coalition are failed in their mission. Also it must be noted that the feasibility of path planning can

not be regarded as a factor in the decision making. It is because path planning is a time consuming and complex activity. In this system to leverage the impact of this problem the mentioned workflow is changed in two steps.

The first change is in the implementation phase and it is in path planning. The implemented algorithm in path planning uses the same raw data that have been extracted for decision making in the analysis phase. Hence, without direct invocation of the path planning algorithm in decision making, it can not be guaranteed that a target which will fail to reach in path planning will not be selected in decision making. Also the selected paths using this method are more reliable than those extracted by means of classic algorithms and other proposed methods.

The second change is in the architecture of the agent's workflow. Although the strategies mentioned above play an important role in increasing the confidence of finding reachable targets and hence decreasing the possibility of loosing cycles, but still loosing a working cycle could be a huge burden over the system's performance and may result in malfunctioning of agents. Therefore some changes have been made to the agent's workflow. This change is in form of adding a feedback from the implementation layer to the decision making layer (see fig. 2). This way after selecting a target, it would be routed and the result would be evaluated. If the extracted path was applicable, then the decision would be implemented. Otherwise the selected building would be removed from candidate buildings and the agent would re-decide. According to the experiments, since the buildings are spread all over the city, usually in the 300 working cycle of the agents, only the agents needed to re-decide more than three times for 5-10 cycles. Also they managed to find a well suited target by two times of re-decision.

4 Police Force Agents

In Rescue Simulation, the police forces are supposed to clear roads. Trying to clear more roads is not the optimum action. Polices have to select the most important roads. Importance of a road is defined as how many times other agents will pass through this road in the following cycles. Although the exact value of this measure is not computable before the simulation, the probable value is computed. For this purpose, roads are prioritized according to two types of parameters, dependant to the system state, and independent of the current state of the system.

Static value of roads: It is obvious that some roads like those around the city are less important than the central roads. To estimate this kind of importance, it is assumed the roads that are subject to use more frequently are more important. We compute this value in this way:

1. The city is saved in form of graph so that the nodes are the graph's vertexes and the roads are the edges.
2. Taking advantage of Floyd's shortest path, all shortest paths between all pairwise nodes are computed and saved.

3. For each edge (road), the number of occurrence of that road in shortest paths are counted.
4. The static value of each road is equal to the number of occurrence of this road in all the shortest paths. This way the road that is more reliable to appear in a selected path is more important.

Dynamic value for roads: At first all dynamic road values are equal to one. During the simulation if an agents requests for clearing a road, its priority is multiplied by ten. The roads around the burning buildings are also multiplied by two. As time passes this dynamic value for each road is decreased.

Total roads values: It is computed as the road's dynamic value times its static value.

This roads values are considered when the police has decided what to do according to his state in the disaster. In order to improve the police forces' in decision making, reinforcement learning has been used. In this method each agent has three actions as follows:

- Stay in his area: When this action is chosen, the police stays in his area and walks around the area he is currently in.
- Help other agents: Selection of this action means the police will be leaving his current area so as to clear a specific road to help another agent achieve his goal
- Change the area of responsibility

Q-Learning[10] has been used for this learning. In latter version of Arian's police forces an algorithm based on function approximation has been used. More details could be found in [1].

5 Ambulance Teams

Ambulance team agents rescue injured civilians. They obtain information of civilians by means of communication, and gathering auditory and visual information. In order to determine whether to go to rescue a civilian or move around to find an injured civilian, a QLearning method has been used.

6 Inter-agent Communication

In the inter-agent communication model, each agent can communicate with its homogeneous agents as well as its center. In the implemented communication system there are two kind of messages:

- Informative messages: These are the messages which are aimed for informing the agents. For example police forces inform ambulance teams of the injured civilian positions.
- Imperative messages: These messages are requests for help. For example fire brigades ask police forces to clear a specific road.

One of the challenges each agent faces is determining what to do among his own decisions according to his intelligence and the requests received from other agents. Design and implementation of a useful method for selecting the best messages to evaluate was of the most important challenges the implemented agents had to face. In the suggested algorithm, each agent sets a priority in the message that is dependent on the following factors :

- Goal's importance: the more important the goal of the sender, the more important is his message.
- The importance of the request in reaching the agent's goal.
- The importance of other possible actions the agent can do if not considering his request.

Each agent with his internal algorithm finds a priority for his decision and comparing this priority with the priority of the messages decides what to do.

Since the way the agents' priorities may be different from each other, and from the priorities set in the messages, the agents use a function approximation method[7] so that in case of failure, the way they set priorities would change.

6.1 Message Evaluation

In our communication system every agent who asks for help, sets a priority in that message. This priority indicates the importance of the situation and how accurate the information is.

One of the problems in communication is to derive a value for the message. we derived an equation for evaluating the message, that is:

$$V_i(t) = f(t) * P(i)/succ(t-1)$$

In the above equation $V_i(t)$ indicates the value of i'th message in each time period and of course each agent has its own queue of messages and has its own $V_i(t)$. $f(t)$ is a coefficient which is related to time and so makes $V_i(t)$ related to time. $f(t)$ is what we are aiming to learn and has a constant value at initial state. By learning $f(t)$ we can say that $V_i(t)$ is learned. $P(i)$ indicates the priority that the message has. $succ(t-1)$ is the measure of the success of the agent in last time period. If this value of $f(t)$ is less than a threshold we will accept the message and do what the message has told us to do, if not, we will ignore that message.

We use and incremental function approximation method for learning the value of $f(t)$. In our function approximation method we use an initial value for $f(t)$ and in each cycle we update the $f(t)$ function. The way we update this function has an explicit relation with the agent abilities and the environment of the problem. Further information and more detailed discussion on the above method and the formula can be found in [1] .

Table 1. Result of RoboCup2002 final game

Team Name	Result
Arian	90.46
YowAI2002	87.85

Table 2. Result of RoboCup2002 semi-final game

Team Name	Environment 1	Environment 2	Environment 3	Environment 4
Arian	68.58	75.77	85.94	53.49
NITRescue	22.56	66.56	54.61	88.15

7 Experimental Results and Conclusion

The implemented agents has participated in the RoboCup Rescue Simulation league in the RoboCup competitions. For the Seattle-2001 competitions, the Arian team succeeded to won the silver medal. In the Fukuoka-2002 competitions, the Arian team managed to become the world champion of the league. Undoubtedly achieving such positions, specially the golden medal in 2002 competitions were the result of the methods and algorithms that we discussed in this paper.

The way participating teams are evaluated against each other, is based on the number of dead civilians, and the number of burned buildings. In the table 1,2 a comparison of Arian score and other final and semi-final teams's scores is reflected [3]. It must be mentioned in the competitions the cooperation and coordination between Arian agents were very dominant.

References

1. Mazda Ahmadi, Mayssam Sayyadian, Jafar Habibi, A Learning Method for Evaluating Messages in Multi Agent Systems, Proceedings of the Agent Communication Languages (ACL2002) , Italy, Bolognia
2. A. Deshpande, D. N. Godbole, A. G. and P. Varaiya, "Design and Evaluation Tools for Automated Highway Systems", Hybrid Systems pp. 138-148, 1995.
3. Luke Hunsberger, Barbara J. Grosz, A Combinatorial Auction for Collaborative Planning, Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000), pages 151–158, 2000
4. H. Kitano, S. Tadokor, H. Noda, I Matsubara, T. Takhasi, A. Shinjou, and S. Shimada. Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research. In Proc. of the IEEE Conference on Systems, Men, and Cybernetics, 1999.
5. I. Noda and P. Stone, "The RoboCup Soccer Server and CMUnited Clients: Implemented Infrastructure for MAS Research" Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publisher, 2002.
6. "http://www.robocup.org"
7. Van Roy, B., Learning and Value Function Approximation in Complex Decision Processes, Ph.D. Thesis, Massachusetts Institute of Technology, May 1998. 30

[3] This score is computed based on how much the agents have reduced the city damage.

8. T. Sandholm and V. Lesser, "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework", Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95), The MIT Press: Cambridge, MA, USA, San Francisco, CA, USA, pp. 328–335, 1995.
9. D. Simon, B. Espiau, E. Castillo and K. Kapellos, "Computer-aided design of a generic robot controller handling reactivity and real-time control issues", RR-1801, 1995.
10. Christopher J. Watkins and Peter Dayan. Q-learning. Machine Learning, 8:279–292, 1992.
11. Edo Zurel, Noam Nisan, An Efficient Approximate Allocation Algorithm for Combinatorial Auctions, Proceedings of ACM Conference on Electronic Commerce (EC-2001), 2001