Impossibles Team Description

Jafar Habibi Alireza Fathi Saeed Hassanpour
Mohammadreza Ghodsi Behzad Sadjadi
Hamidreza Vaezi Majid Valipour
Sharif University of Technology
Computer Engineering
IRAN, Tehran

February 22, 2005

Abstract

In this paper we try to describe the main strategy and algorithms implemented in Robocup Rescue agents built by Impossible team at Sharif University of technology. We have made a base layer by restructuring the ADK source code which make our agent algorithms independent from the lower level codes, easing the implementation of agent controllers. Some algorithmic strategies such as Cooperation & auction, Reinforcement learning, probabilistic modeling are briefly described.

1 Introduction

Recently, multi-agent systems have gained a lot of attention. Robotic soccer[1], automated driving and traffic light control[2]. RoboCupRescue Simulation with heterogeneous agents with different abilities and of course responsibilities with a limited communication, is an excellent framework for multi-agent planning, communication techniques, coalition formation and task allocation. So we tried to improve the algorithms for this task with hope that these algorithms can be useful in the similar situations in the real word. In the Figure 1 our base layer hierarchy which are agents are implemented on this base is shown.

2 Memory Structure using Probabilistic World Modeling

In multi agent systems, each agent gets its information from the outer world and other agents by either sensing or communication. Agents maintain a data

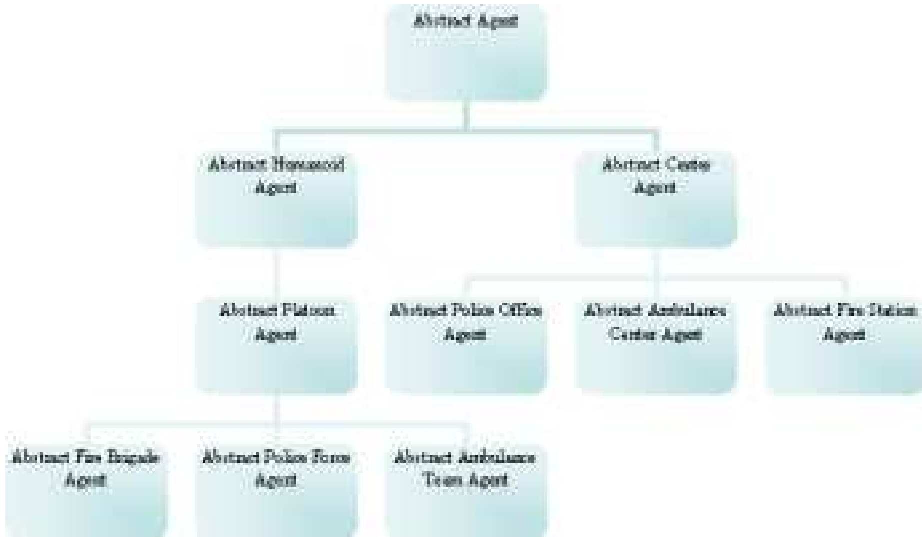

Figure 1: Base layer Hierarchy Diagram

structure which is called the world model. In Robocup Rescue Simulation library, there is a simple world model class, which has a list of objects existing in the disaster space and also their properties. Obviously this simple world model is not efficient for our complicated and dynamic world.

In the real world, we have two kinds of values assigned to objects, the first one is set of values which we know for sure, and have their precise values. The second category is values whose exact values are unknown and change in every cycle as disaster space changes. Our probabilistic world model (PWM) concentrates on the latter category for decision making.

To fulfill this objective we obtain probability distribution functions (pdf) for each property in the second category and estimate their values in every cycle. These estimated values are used whenever needed in decision making algorithms. This concept has close relationship with fuzzy logic and we can use fuzzy methods on this model in future implementations.

3 Communication Structure

Theoretically we can implement a distributed multi-agent system without any communication between agents. But in practice we need communication to transfer information between agents to make decision process easy. In the best case if the message length was more than the length of sense incoming message, each agent could have enough information to decide independently without lacking available data sensed by other agents. Since there is a limit on message lengths. We have to use a priority deciding policy to choose the most important and essential messages to send. Also we can save the lower priority messages to be sent in coming cycles. We have a centralized priority setting mechanism in center agents to manage message passing in team agents.

4 Decision Making

4.1 Auction

In auctions the outcome is usually a deal between two agents:the auctioneer and one bidder. An auction consists of an auctioneer and potential bidders. Auctions are usually discussed in situations where the auctioneer wants to sell an item and get the highest possible payment for it while the bidders want to acquire the item at the lowest possible price.

We have used auction algorithm for our fire brigade agents. Fire brigade agents are auctioneers who want to sell themselves and the ignited building blocks are bidders. The distance between building blocks and agents, size of the burning area and the time fire spread started are the most important characters affecting the money a building block as a bidder has.

4.2 Reinforcement learning

In reinforcement learning reactive and adaptive agents are given a description of the current state and have to choose the next action from a set of possible actions so as to maximize a scalar reinforcement or feedback received after each action.

We have used reinforcement learning for police forces in different maps and also used for setting the priority of roads in each map.

4.3 Path Finding

We have used various algorithms for path finding in the city. We have used BFS algorithm for police forces and Dijkstra algorithm for ambulance teams and fire brigades. The main point in the Dijkstra algorithm is giving weight to edges of the graph. At the beginning we tried to give weights to edges by examining different graph states. But then we used reinforcement learning to define the priority of roads and giving weight to them.

5 Conclusion and Future Work

We have reached this conclusion that there should be a trade off between heuristic and learning methods in such complicated environment. Sometimes simple heuristic methods get better results than complicated learning algorithms and vice versa. We are planning to implement four methods: Fuzzy logic, Reinforcement learning, cooperation and neural networks in our agents.

References

[1] 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

[2] Macro Wiering, Jelle van Veenen, Jilles Vreeken, Arne Koopman; *Intelligence Traffic Light Control*; (2004)

[3] Gerhard Weiss; *Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence*; The MIT Press, 1999

[4] George F.Luger, Williams A.Stubblefield; *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* Addison Wesley Longman, 1998

[5] Richard S.Sutton, Andrew G.Barto; *Reinforcement Learning* (2002)

[6] Tom M.Mitchell; *Machine Learning* McGraw-Hill, 1997