The common knowledge model of a team of rescue agents

M.L. Fassaert, S.B.M Post, and A. Visser
Computer Science Department, University of Amsterdam, Kruislaan 403,
1098 SJ Amsterdam, the Netherlands
Tel: ++ 31 20 525 7532 - Fax: ++31 20 525 7490
Email: mfassaer,sbmpost,arnoud@science.uva.nl
http://www.science.uva.nl/˜arnoud/research/roboresc/

***Abstract*— In this article we will concentrate on the underlying knowledge model used in the design of a multi-agent system, operating within the 'RoboCupRescue' Simulator system. To cope with the limited communication in this application we introduce teams of cooperating agents that build a common knowledge model. Based on the common knowledge in the teams model, the agents are able to predict the behavior of their teammates. This opens the possibility to cooperate without explicit communication. As basis of the common knowledge model we use sectors, a representation that is worked out in more detail in this article.**

I. Introduction

After participating in the Soccer Competition since 1998 [1], the University of Amsterdam extends its interest in 'Rescue Simulation League' [2] with the team 'UvA-Rescue C2003'. The 'C' in our teamname stands for 'Communication', because the core of our approach is to reduce the communication for multiagent coordination [3]. The 'RoboCupRescue Simulator System' (RCRSS) simulates a small piece of a real world environment, in which a disaster takes place. The simulation starts with an earthquake, after which several buildings in the disaster map collapse. This causes buildings to catch fire, roads to be blocked and civilians to get buried under the debris. These events affect the agents on the map and their goal is to respond in the most appropriate way, in order to minimize damage. Platoon agents perform the rescue tasks in the field, orchestrated by a center for each operational type. The firebrigade extinguishes fires, ambulance units rescue civilians from under the debris and transport them to refuges, and police units clear the roads. Typically around twenty-five platoon agents are available in a competition simulation.

The platoon agents require a fair amount of localized intelligence. There are many practical problems to overcome like getting to the location of the next task, having enough water to put out the fire and making sure there is room in an ambulance to pick up more wounded. With the very limited communication between the platoons and the centers it is vital that not every detail of a task needs to be communicated to the planning agents. Even some coordination with homogenous and heterogenous units in the neighborhood should be done autonomously, especially since communication through the center agents clogs up the communication line between them.

II. Communication and Coordination

The coordinating part, both locally and globally, relies heavily on information gathered in the field. The platoon agents are not only the hands of the centers but also their eyes and ears. Because of the limited visual range of agents, information gathering will be an active task. A part of the available platoons should be patrolling the disaster space looking for problems that are not known to the team and control centers. After a risk area is identified, a risk assessment is needed for local and global decision making. This will require the measuring of the extends of a fire, the number of collapsed buildings, the number of buried civilians and finally finding of routes to the problem area.

Agents translate the information gathered in the field to a world model that is usable by any form of a decision making process and is as good a mapping of the actual situation as possible. The worldmodel has to be based on summaries of what agents encounter. The level of compression of those summaries has to be a compromise between the detail needed for decisions and the overview needed to make this decisions in time.

The centers return the collected information to the agents allowing them to form teams and choose a task to focus on. Besides active information gathering, these tasks can be extinguishing a group of burning buildings, evacuating the civilians from a high risk area or improving the roads between refuges and danger areas. The members of a team work together in performing a task and keep the centers updated on the progress.

We will introduce the concept of sectors since they are the basis of our world model. They are used in both communcia-tion and coordination. Then we will briefly describe how our decision making process and communication process is going to work and how the use of sectors improves both.

III. Sectors

A sector is a subset of the non-moving objects on the map. Sectors can be predefined for a certain city by the civil security organisations, or dynamically generated for an unknown (virtual) map of a city. Currently we are using an algorithm that automatically generates sectors just before an agent connects to the kernel. The map is divided in sectors by applying a grid of regulary distributed nodes over the map as potential cornerpoints of the sectors. The shortest paths between those cornerpoints are used as boundaries of the sectors. All non-moving objects inside these boundaries as well as the boundaries themselves are part of the sector. This ensures that all items within a sector are reachable from every other item in that sector without traveling over items that are not part of the sector. During simulation this cannot be guaranteed due to road-blocks, which are only discovered after initialization. In that case we allow detours over the sector-boundaries to be part of the knowledge about the sector. This requirement is needed so an agent can use the sector as a consistent subset of the worldmodel to compute actions for.

Fig. 1. A distribution of 5x5 sectors over the Kobe map

Inside a sector there will be several streets. Streets are in our terminology a sequence of roads without branches or crossings. Routes are sequences of streets where the begin and end-point can be found on the boundary of a sector. Routes can be used to travel long distances, from one sector to another. There is a route to travel from each sector to every other sector. The limited number of objects inside a sector also makes it viable to precompute paths between all the objects inside the sector. By varying the number of sectors used we can optimize the time and memory required to create inter-sector paths and intra-sector routes. In fig. 2, for example, one can see that the optimum for the Kobe map can be found at 5x5 sectors.

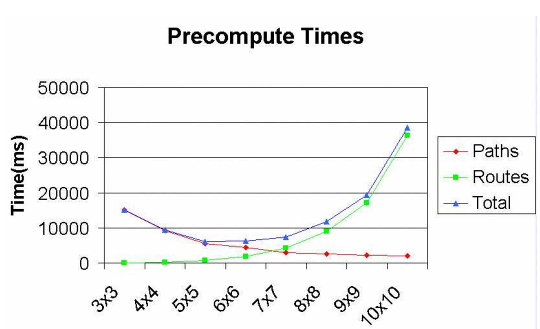

Fig. 2. The computation time used to divide the Kobe map in different numbers of sectors.

IV. Agent Behavior

For the agents we have created a library of behaviors where they can choose the most appropiate one based on their view on the current situation. So far the library consists of 9 behaviors.

DeblockRefuge	Clear a route between the current sector and a refuge.
DeblockRoute	Clear a route from one sector to another
DeblockSector	Clear the important streets inside a sector.
DeblockTeammember	Clear a path to the location of a teammember
ExtinguishBuilding	Extinguish a burning building
RefillWater	Go to a refuge and refill the water-tank
SafeCivilian	Rescue a civilian from a collapsed building
SafeTeammember	Rescue a platoon agent from a collapsed building
EvacuateVictim	Load a civilian and bring him to a refuge
Evacuate	Move from a burning house to the nearest road
Patrol	Move around and search for fire, victims or road-blocks

Currently one of those behaviors is selected based on the agents capabilities, and three criteria:

Priority	this value differs for the different behaviors. Currently we like to give for instance a higher priority to DeblockTeamMember then to DeblockRefuge.
Attainability	this value is higher when there is a greater chance that the behavior will be executed succesfully.
Reachability	this value is higher when the travel time to the target location of the behavior is lower.

Eventually we will extend this selection process, taking into account the behaviors of other agents. Our solution will be based on the research performed for soccer agents [4]. One of the applications of the sectors described earlier will be to reduce the rescue problem to several smaller problems. This greatly reduces the number of considered behaviors. In combination with the precomputed paths, needed for the reachability, the time used when selecting behaviors is so low that it allows agents to predict the behavior of other agents. By predicting the behaviors of other agents they are able cooperate better without using communication. The fast behavior selection even makes it possible to predict the effect of a sequence of executed behaviors on the future situation in a game-tree like fashion.

In fig. 3 we demonstrate this for a single police-agent, making decisions based on its own perceptions. With our approach the police-agent needs half of the time needed by the classical A* approach to select an appropriate behavior for

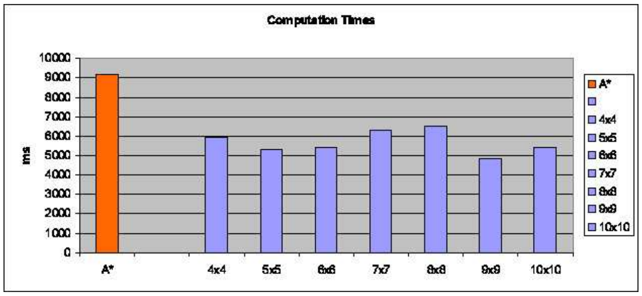

Fig. 3. The total computing time used to select the best action for one policeagent clearing roads in Kobe over 300 timesteps. The A* time is the time used if precomputed paths and sectors are disabled.

himself. This leaves enough computing time to estimate the appropriate behavior of about ten other agents in that cycle (for a kernel time-step of 2 seconds, with two 1-GHz workstations for the agents). We expect the advantage to be even more apparent when the policeagent has to pick from targets that he is told about via the communication channel. Those targets will be located at larger distance, so in the selection process one has to consider traveling further and will benefit more from this optimization. Ambulances and fireagents will benefit even more, since they tend to travel longer distances then police-agents for their tasks.

V. Communication

Our second goal was making sure that the information provided by the agents in the field is precise and fast enough to make decisions on a global scale by the centers. We will do this by translating the information gathered in the field to a world model that is usable by any form of a decision making process and is as good a mapping of the actual situation as possible. The worldmodel is based on summaries of what agents encounter in the field so it will not be a perfectly detailed model, nor should it be. The second use for the earlier described sectors will be to create a language for these summaries that reduces the temporal and spatial resolution of the world model.

To measure this we have to compare the combined world-models of all agents in the field with the summarized world-model in the centers and count the number of differences at every iteration. It is expected that the error due to communication lag will be large in the first few iterations of the simulation because the difference between the global worldmodel of the center the local worldmodel of the fieldagents will be large. When this difference becomes less the communication lines will become more readily available and the error in the center's worldmodel will asymptotically decrease.

VI. Conclusion

We have analyzed problem areas in the design of a multi-agent system for the RoboCupRescue simulator environment and we have defined the coordination and communication required to perform the rescue task well. We have proposed the use of sectors to lower the complexity of the problem and described how this will aid the overall performance by reducing the use of the limited resources. With this research we will cooperate in this year competition on this area, the RoboCup Rescue Simulation League.

Acknowledgements

This paper is based on preliminary research of the RCRSS and ADK done by: S.B.M. Post, K. Luikx, J. Uijlings, R. Geertzema and V. Mahabir.

References

[1] E. Corten and F. Groen. Team description of the UvA Team. In RoboCup-99 Team Descriptions for the Simulation League, pages 149-153. Linkoping University Press, 1999.

[2] T. Takahashi and S. Tadokoro Working with Robots in Disasters - RoboCup Rescue: A Multi-Agent Approach to Disaster Simulation. In IEEE Robotics & Automation Magazine, p. 34-39, September 2002.

[3] S.Post, M. Fassaert, A. Visser "Reducing the communication for multi-agent coordination in the RoboCupRescue Simulator", submitted to the 7th RoboCup International Symposium, Padua, Italy, 10-11 July 2 003..

[4] J. Kok, M. Spaan, and N. Vlassis. An approach to noncommunicative multiagent coordination in continuous domains, in Benelearn 2002: Proceedings of the Twelfth Belgian-Dutch Conference on Machine Learning, pages 46-52, Utrecht, The Netherlands, December 2002.