# The high-level communication model for multi-agent coordination in the RoboCupRescue Simulator

S.B.M. Post, M.L. Fassaert, 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
{sbmpost, mfassaer, arnoud}@science.uva.nl
http://www.science.uva.nl/~arnoud/research/roboresc/

**Abstract.** In this article we will concentrate on the communication problems in a multi-agent system, operating within the 'RoboCupRescue' Simulator system. To cope with the limited communication between the center and the agents in the field, we separate the communication in two layers that focus on synchronizing world models with different levels of detail, responsiveness and range. In this article we will explain the requirements and methods used in the high-level communication that distributes summaries of the current situation in different sectors of the map.

## 1 Introduction

The world is often shocked by catastrophic events. These may range from natural occurrences, such as earthquakes or floods, to urban riots or terrorist attacks. Crisis management is essential under such circumstances. Unfortunately, real large catastrophes do not only mean thousands of deaths or injured people, but also hit the communication and civil protection infrastructure, jeopardizing all but the most carefully prepared rescue plans. There is always more to be done in order to minimize such damage in the future. It is possible that the new solutions may be aided by the use of advanced technologies such as robotic rescue operators and artificially intelligent expert systems.

In this article we will investigate multi-agent communication within the 'Robo CupRescue Simulator System' [1]. The 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. Simulated rescue agents need to respond in the most appropriate way, in order to minimize damage.

In section 2 we will determine what kind of information system is needed for the agents to accomplish this goal. In section 3 we will then explain the details of how we implemented this system. Finally in section 4 we will show that our design does indeed meet our requirements.

# 2 Requirements

## 2.1 Agent Objectives

There are three different types of intelligent entities in the simulator system: civilians, agents, and centers. Civilians represent ordinary people who can not be controlled by our multi-agent system, they are programmed to flee from danger and run to refuge buildings. The main objective of agents and centers is to save as many civilians as possible and to minimize damage due to fires. The agents and centers are separated in three operational types: the fire brigade to put out fires, ambulance units to rescue civilians from under the debris and to transport them to refuges, and police units to clear the roads.

The agents perform the actual rescue tasks. This requires a fair amount of localized intelligence. There are many practical problems to overcome like getting to the location of the next task, making sure a fire brigade has enough water to put out the fire and making sure there is room in an ambulance to pick up more wounded. Simple operational subprograms take care of the most basic logistics. We call these subprograms 'behaviors'. Selecting which behavior to execute is a decision making process that can be handled in many different ways. In our approach all the decision making is done by the localized intelligence of the agent. The suggested hierarchy of the centers standing above the agents is used only for spreading information and not for coordinating task assignment. The agents decide which other agents to cooperate with and what tasks to perform. To do this they need an overview of the global situation. They can not base this overview on their limited visual observations alone. This is why communication is required to update this overview.

## 2.2 Situation Awareness

The coordinating part of each agent relies heavily on information. Because of the limited visual range of agents, information gathering will be an active task. Some of the available agents will be patrolling the disaster space looking for problems that are not yet known by the agents. After a risk area is identified, a risk assessment is made. This consists of the measuring of the extends of a fire, the number of collapsed buildings, the number of buried civilians and their survival chance and finally the number of blocked roads. Based on this information a distribution of agents to tasks can be made.

The information that is observed by an agent is combined with the information that is communicated to it into a world model [2], which is used in the decision making process. This world model is based on summaries of what agents encounter in the field so it will not be a perfectly detailed model, nor should it be. Cutting down on the number of variables in the world will make the job of the decision making process easier. It is also a more realistic simulation of what goes on in a real disaster control center.

The information in the summaries can be limited by aggregating the details for certain areas. By grouping the elements on the map (roads, building) of the simulated city by a algorithm described in [?], the information that has to be exchanged can reduced considerably. We call the grouped elements on the map sectors. A sector can be seen as a possible problem area. Observations of properties like burning buildings, blocked roads, collapsed buildings can be summed up to give a factor of these problems for all sectors. The positions of all agents are reported and send along with the sector values. Together these values form an information package, that is sufficient for an agent to decide which problem area to go to. By making this decision for other agents as well, an agent can find out what other agents are going to be aiding it there.

The information in the summaries does not need to be detailed in the sense that agents can plan their actions with it. For this the agents use their observations supplement by a more direct method of communication. The execution of the tasks and the communication required for this is not the focus of this paper. The coordination of what problem area to focus on is separated from the cooperation that is required to fulfill this task.

### 2.3 Information usage

The right number of agents should cooperate to accomplish a task. Too many agents dealing with a single fire is wasteful and hampers secondary goals like patrolling or containment of another fire. A decent estimation of the required team composition has to be made. To do this the information in the world model needs to be accurate.

The precision of the summaries is best secured by making enough observations. Some agents are reserved to patrol the area and make these observations. The combination of these observations into one state vector has to be done by weighing new data based on trustworthiness. In this process visual observations are trusted over communicated ones. The time of observations is remembered and compared with the time of arrived communication. Only communicated information that is newer than the last observation should be integrated in the overview. Only the observations that were made since the last communication, should be transmitted to other agents.

Agents use the world model to choose their next task. Because of the limited time and number of agents during a simulation, this has to be done in a very efficient way. It is important that fire agents travel towards a new fire as soon as it has been detected. Because of this the information in the world model has to be recent. On the other hand big decisions like switching teams do not have to be reconsidered as often as decisions on what action to take next. A reaction to every small change in the overview would make it harder for agents to finish their tasks. This is why agents choose new tasks and teams only after new global information is available and therefore this information does not have to be updated every cycle. As an extra bonus combining the data of sectors over a period of time, further reduces the amount of information transmitted concerning overviews by discretizing it in the time dimension. Still the information needs to reflect sudden developments accurately, so the period we chose for these updates is the minimum number of cycles required to synchronize the summaries between agents.

To prevent misunderstandings in the cooperation of agents the information in the agents needs to match. The centers not only fulfill this requirement by distributing the information, but also an active role in maintaining a common view. They accumulate observations from the agents they have a direct communication channel with, which are agents of the same type. Next they combine these into one summary by negotiating with each other. The same common summary is then send back to all agents at the same time. This guarantees that all agents base their task and team decisions on the same information.

## 3 Solution

This section will explain the working of the communication protocol which is an attempt to make the world model as accurate as possible. The protocol consists of two different communication layers. In the low level communication layer, 'low level information' is transferred between agents that are near to each other. The high level communication layer distributes 'high level information' amongst all of the platoon and center agents. Low level information refers to basic observations that are directly relayed to other agents in the field, whereas high level information is first extracted from the low level information before it is distributed. The two communication layers match nicely on the two different kinds of message primitives provided by the kernel.

The message primitives are implemented by means of SAY and TELL messages. SAY messages can be used to talk to another agent and it only reaches agents in a limited range. However it does not matter what kind of agent is on the receiving side. The opposite is true for TELL messages where agents must be of the same type in order to communicate, but it arrives at all agents and centers of the same type regardless of distance. For both message primitives certain restrictions are imposed on the amount of messages that may be sent or received during each simulation cycle. Moreover each message should adhere to a special format. The details have been summarized in tables 1 and 2.

| AGENT CATEGORY | AK_TELL/AK_SAY | KA_HEAR |
|---|---|---|
| Platoon agent | 4 | 4 |
| Center agent | $2 * n$ | $2 * n$ |

**Table 1.** Agents discard messages they do not want to hear, using only sender and receiver identities contained within the message; Tell messages can only be heard by agents of the same type, i.e an ambulance-center that controls ambulance agents, where $n$ is the number of platoon agents that are in control of a center; Say messages can only be heard by agents that are within a $30m$ radius while the visual range is $10m$.

| | | | |
|---|---|---|---|
| sending: | SENDER_ID | MESSAGE | |
| receiving: | RECEIVER_ID | SENDER_ID | MESSAGE |

**Table 2.** Message format for agents; sending relates to AK_TELL and AK_SAY messages, whereas receiving relates to KA_HEAR messages. Note that the message size is limited to 80 bytes.

### 3.1 The communication protocol

Within the communication protocol TELL messages are used for the high level communication layer and SAY messages for the low level communication layer. The presence of communication restrictions during each cycle leads to the definition of 'communication phases'. An agent can only enter a new phase after at least one cycle has passed since the previous phase was entered. For high level communication, each phase has the property not to violate the restrictions that were mentioned in previous section. For example platoon agents will not produce more than four messages during a phase. It is also ensured that the receiving party is ready to accept all incoming messages without the need to throw them away. In particular centers indicate that they are ready for reception by sending a single TELL message. Thus within the high level layer, agents wait until all participating agents have reached the appropriate phase before they exchange information. The following phases have been defined:

- Platoon agents: AC_EXCHANGE, AA_SAY
- Centers: AC_EXCHANGE, CC_READY, CC_EXCHANGE, CA_READY

These phases are entered in this specific order (reading from left to right) and when the centers enter the last phase, one high level communication cycle has finished. Then a new communication cycle starts and each agent enters the first phase again. Experiments showed that 5 kernel cycles were needed to finish 1 high level communication cycle. So during the simulation there are $300/5 = 60$ high level cycles.

The first two letters of each phase describe what kind of agents participate in the communication. For example the prefix AC means that there is communication between a platoon agent and a center agent. The suffix gives an indication of what happens during a particular phase. The following list clarifies this:

- AC_EXCHANGE
  *Platoon agents provide their centers with new world information, after which they obtain summaries that have been produced by the centers. Soon after, the platoon agents enter the AA_SAY phase, enabling them to do low level communication.*
- CC_READY
  *The centers prepare to enter the CC_EXCHANGE phase by first entering the CC_READY phase. In this phase they notify other centers that they are ready. This is when all AC_EXCHANGE messages have been processed.*

- CC_EXCHANGE
  *The centers distribute the new world information among each other, and merge the into summaries. Then they enter the CA_READY phase.*
- CA_READY
  *Platoon agents prepare to enter the AC_EXCHANGE phase, when their centers notify that they are ready.*

So what actually happens is that agents communicate with their centers, exchanging information. Then the centers communicate with each other while the platoon agents can do low level communication with agents in their vicinity. Once the centers are ready the process starts over again. We want to emphasize that this high level communication scheme is synchronous; all agents know about the same information at approximately the same time. This is very useful because by using high level information (summaries) one can predict what other agents are doing. Another feature is that if for some reason the high level communication layer breaks, then the platoon agents remain in the AA_SAY state so the low level layer is still operational. Table 3 summarizes the differences between the high level and the low level communication layer.

| property | HIGH LEVEL | LOW LEVEL |
|---|---|---|
| datastructure used : | summaries | object data |
| communication pattern : | platoon agents to centers and back | only between platoon agents |
| agent types : | homogeneous | heterogeneous |
| message primitive : | TELL messages | SAY messages |
| reception : | synchronous, nothing lost | uncertain |

**Table 3.** High level communication versus low level communication. A summary contains high level information, whereas object data equals low level information.

# 4 Research Methods

## 4.1 Proving the Design Goals

More important than our ranking in the RoboCup Rescue competition is proving that we reached our design goal presented in this document. Our goal was making sure that the information provided by the proposed communication system is accurate, and recent enough to base decisions on and lastly that it matches whenever it is used to make decisions. We have measured the total percentages of buildings that are known to be burning, during a simulation that is typical for a competition. The map we used is known is Kobe and it is measured over 300 cycles with a couple of stationary agents on different sides of the map.

### 4.2 Results

To measure the precision we have to compare the world models of an agent in the field with the actual situation. It is expected that the error will be large in the first few iterations. As agents scout out the map their approximation of the real situation should improve fast and keep up with new developments like spreading fire. The amount of spreading fire as known to the kernel is shown in figure 1. This amount is compared to the knowledge of agents. We have measured the amount of fire known by two agents and show that it exceeds the amount of fire known by a single agent.

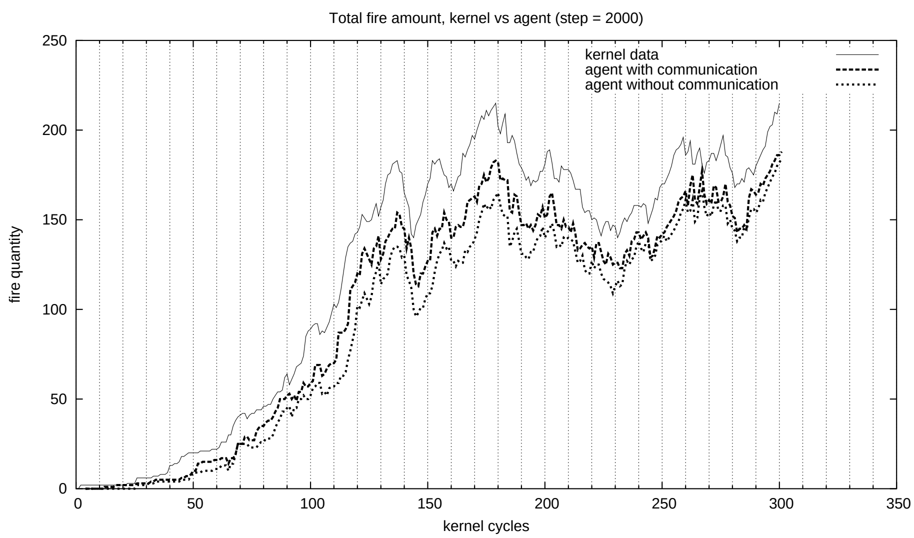


**Fig. 1.** Two agents patrolling together increase their knowledge of burning buildings over that of a single patrolling agent. The vertical lines mark high level communication cycles

The amount of known fire of an agent in communication with another agent exceeds the amount of known fire of a single agent. What is more important is that the error between the amount of actual fire and the amount of known fire is a lot smaller. Because large, hard to extinguish fires are easier to spot from far away the difference mostly consists of young fires that are easier to extinguish and therefore more important to take action against.

## 5 Conclusion

With an early version of this approach we have competed in the 2003 version of RoboCup Rescue Simulation League [3]. Unfortunately technical problems and flaws in our situation negotiation algorithm prevented us from seeing the effects of our approach reflected in our competition scores. We have shown that it is now working as expected and hope to see improvements in our competition results because of it.

## 6 Acknowledgments

This paper is based on preliminary research on RoboRescue Simulation done by: S.Post, K.Luikx, J.Uijlings, R.Geertzema and V.Mahabir

## References

1. T. Morimoto `http://ne.cs.uec.ac.jp/~morimoto/rescue/manual/manual-0_00alpha1.ps.gz`
2. M.L. Fassaert, S.B.M. Post, A. Visser "The common knowledge model of a team of rescue agents", 1th International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster, Padova, Italy, 6 July 2003
3. A. Visser, S.B.M. Post, M.L. Fassaert, "The communication approach of the 'UvA Rescue C2003'-team", in D. Polani, B. Browning, A. Bonarini, K. Yoshida (Eds.), RoboCup 2003, Lecture Notes on Artificial Intelligence, Springer Verlag, Berlin.