LTI Agent Rescue Team: A BDI-based Approach for Robocup Rescue

André H. Pereira, Luis G. Nardin,
Anarosa A. F. Brandão and Jaime S. Sichman

Laboratório de Técnicas Inteligentes (LTI)
Escola Politécnica (EP)
Universidade de São Paulo (USP)
Av. Prof. Luciano Gualberto, 158 – trav. 3
05508-970 – São Paulo – SP – Brasil
{andre.hahn,luis.nardin}@usp.br,
{anarosa.brandao,jaime.sichman}@poli.usp.br

Abstract. In this paper, we propose the use of a BDI agent architecture to implement a team for the Robocup Rescue simulation. This approach takes advantage of both local and global information to improve coordination and task allocation among platoon agents, that may act independently from their center agents. Therefore, in our approach, center agents do not coordinate directly platoon agents. On the other hand, since center agents have a broader and consolidated view of the environment, they may provide optimized information for platoon agents, which hence may benefit from global information when necessary.

1 Introduction

Coordination is one of the key issues in cooperative multiagent systems (MAS) since it ensures that agents will not accidentally interfere with each other's subgoals while attempting to achieve a common goal [10]. Therefore, coordination can be defined as the act of handling the inter-dependencies between the activities of the agents in order to maintain the system coherence.

Disaster may be defined as a crisis situation causing wide spread damage which far exceeds our recovery ability. Its management is normally critical, since it involves a very large number of heterogeneous entities in a complex and dynamic environment, where the information is incomplete and uncertain, and decisions must be taken in a timely fashion. In those situations, coordination and cooperation play an essential role in the assistance provisioning. The Robocup Rescue Agent Simulation[1] competition promotes the research and development of efficient response policies to such disaster scenarios, which involves scoring competing multiagent teams' coordination schemes.

In this paper, we propose a task allocation approach, which is based on *Partial Global Planning* [4], to develop the strategies to coordinate rescue agents in

[1] http://www.robocuprescue.org

the Robocup Rescue simulation environment. Such proposed approach, named *Partial Global* approach, considers and handles both local and global information. In our framework, rescue agents are divided into two classes: *platoon agents*, that effectively handle the rescue tasks, and *center agents*, whose goal is just to centralize global information and deliver this information when needed to platoon agents. These two classes are applied to all sort of Robocup Rescue agents: Fire Brigade, Police Force and Ambulance Team agents. Therefore, while platoon agents keep only local information about the environment, center agents consolidate the environment information received from the former. Furthermore, platoon agents prefer to take decisions and act using their local information, requesting information to center agents only when they have no available local information and they are idle. Our main idea is that center agents should not directly coordinate platoon agents, but allow the latter to take advantage of global information in order to direct their actions when needed.

In order to implement the rescue agents with the proposed task allocation approach, we propose the use of a *Belief - Desire - Intention* (BDI) architecture [8]. This agent architecture was inspired by and based on a model of human behavior that was developed by philosophers, and it enables us to describe the rescue agents strategies as *mental states.*

The rest of the document is structured as follows. Section 2 briefly describes the BDI architecture and its integration to the simulation environment. The *Partial Global* approach proposed to implement the rescue agents is described in section 3. In section 4, some preliminary results of our approach are presented. Finally, we conclude and provide some future work directions in section 5.

2 Agents Architecture

The *Belief - Desire - Intention* (BDI) paradigm was originated in the work of the Rational Agency project at Stanford Research Institute. It lies on the theory of human practical reasoning developed by Bratman [2]. The BDI architecture views the system as a rational agent having certain *mental attitudes* of Belief, Desire and Intention, representing, respectively, the information, motivational, and deliberative states of the agent. These mental attitudes determine the system's behavior and are critical for achieving adequate or optimal performance when deliberation is subject to resource bounds [9].

In this work, we use *Jason* [1] to implement BDI-based agents. *Jason* is a Java-based platform for the development of multi-agent systems. At the core of the platform lies an interpreter for an extended version of AgentSpeak [7]. AgentSpeak is a logical programming language based on a restricted first-order language with events and actions and it was first developed to bridge the gap between the theory and practice of BDI architectures.

Jason systems are composed of an *Environment*, a *Belief Base* and a *Plans Library.* The *Environment* represents the topological space where the agent evolves and interacts. It restricts what the agent can sense and what actions the agent can perform; its representation depends on the application domain. The *Belief*

Base contains all the beliefs that the agent has about its environment and it consists of a collection of literals, as in traditional logic programming. The *Plans Library* contains all the possible plans an agent may perform; Plans are considered as courses of actions that an agent commits to execute aiming to achieve its goals.

In order to integrate the *Jason* platform to the Robocup Rescue (RCR) simulator it is required to customize and adapt several components:

- We need to implement a *Jason* agent architecture by extending the *AgArch* class to enable the agent to transform the sensed information from RCR into literals in the agents' *Belief Base*. In this work, each sensed information is transformed into one belief in the agents memory, however only the useful information to the agent is stored as a belief;
- We need to implement the available command actions in RCR as *Jason* internal actions by extending the Java class *DefaultInternalAction*;
- We need to implement the proposed *Partial Global* approach and the rescue agents' strategies using the AgentSpeak language.

Despite the fact that *Jason* platform allows direct communication among agents using its own primitives, these are not used in our team since Robocup Rescue does not allow such communication. Due to space limitations, details about Jason are not presented here. Interested readers are pointed to [1].

3 The Partial Global approach

In the Robocup Rescue, reliable information about the environment is vital for efficient action planning. However, since agents have limited sensing and communication capabilities, information sharing is not trivial. Therefore, we decide to use a strategy of centralizing information in the so-called center agents; however, we decide to keep platoon agents autonomous to perform their tasks if they cannot contact their center agents. Our idea is to hold a global and a centralized information repository where all relevant data sensed by the agents are kept. This information is provided by the agents themselves, and then processed and redistributed to all the other agents when requested.

The *Partial Global* approach proposed in this work is based on *Partial Global Planning* [4]. The latter considers that the cooperating agents exchange information in order to reach common conclusions about the problem-solving process [10]. It is considered *partial* because the agents do not have all the environmental information available in order to make the best decision. On the other hand, it is *global* because the agents can exchange and obtain information through the center agents which consolidates environment information known by the agents.

Details about the proposed approach are presented in the next sections. In section 3.1, we describe how rescue agents coordinate among themselves, and in section 3.2, we present the different platoon agents strategies.

3.1 Agents Coordination

The *Partial Global* approach considers that platoon agents can act independently from their center agents, nonetheless they are limited to act based solely on their own local information or on information received from other close agents.

Therefore, in scenarios where platoon agents cannot contact their center agents or the center agents are not available at the simulation, they walk randomly through the environment looking for victims, burning buildings or blockages. Once they find any of these situations, they may act, if they have the required capability, or keep the information in their memory in order to share it with another agent in the future. Hence, the only coordination available in this situation is the information sharing among the agents when they are close together.

However, since in most situations communication with center agents is available, platoon agents behave differently than described above. In such situations, platoon agents begin the simulation walking randomly in the environment looking for victims, burning buildings or blockages. Once they find it, they inform the situation to their center agent and they perform an action, if they have the required capability. If they do not find any situation to act upon after a short period of time, they request information to their center agent. If this latter does not have any information to provide, platoon agents continue to walk randomly for another short period; however, if information is provided, then they calculate the shortest path to the target, they move to the location and start to act on the situation. When the action is finished, they update the current information to their center agents. In the sequence, if platoon agents cannot sense any other situation to act, they request information again and the cycle restarts.

Hence, even if center agents cannot directly coordinate platoon agents, they can perform a sort of indirect coordination when providing information to them. The proposed strategy considers that center agents will receive all information sensed by platoon agents, and based on these information, center agents can determine the priority of actions to be taken by platoon agents and, in this way, coordinate platoon agents' work.

3.2 Agents Strategy

In the Robocup Rescue simulation, an important role in the efficiency of the agents is played by the adopted path planning technique. It is really essential that this technique should be as efficient as possible in order to not impact the other reasoning processes. Therefore, all the agents in this work adopt the Dijkstra's algorithm [3] to deal with shortest path planning to multiple goals.

In the next sections, the strategy of each kind of platoon agent (Fire Brigade Agent, Policy Force Agent and Ambulance Team Agent) is described in more detail.

Fire Brigade Agent In the beginning of the simulation, the fire brigades start by performing the search process. The search process corresponds to the act of searching for fire spots randomly for a short period of time. If a fire brigade finds a fire spot within this short amount of time, it moves to that location to try to extinguish the fire. However, if it cannot find any fire spot within that amount of time, it requests a fire spot location to the fire station.

When the fire brigade gets within the target's fire extinguishing range, it begins to extinguish the fire and it performs this action until either the fire is extinguished or it runs out of water.

In case the fire brigade is unable to reach the target's fire extinguishing range within a predefined amount of time, the target is considered unreachable and it is ignored. Therefore, the fire brigade restarts the search process all over again. For instance, such situation may happen when all the paths to the selected target are blocked and no police force agent was able to clear it on a reasonable time.

When the fire brigade target building has its fire extinguished or it burns completely, the fire brigade sets its target to null, informs the fire station about the situation and restarts the search for a new target.

On the other hand, when a fire brigade runs out of water, it moves to the nearest refuge in order to refill its water tank. Once the fire brigade water tank is filled, it goes back to its last target to resume its firefighting. This means that the agents are persistent with their goals, as predicted by a BDI architecture: they do not change their current goal until it is deemed impossible to reach or the target is no longer a valid goal.

Police Force Agent Clearing paths to refuges is considered crucial in our strategy, since fire brigades need to get to refuges to refill their water tank and ambulance teams need to have fast ways to reach the refuge when transporting civilians. Therefore, the initial objective of all police forces is to clear paths to refuges. Once a police force reaches one refuge, at least one path to that refuge is supposed to exist and the police force resumes its general strategy.

The general strategy of a police force consists of clearing the closest blockage that it has information about, when it is able to perceive any. If it does not know any blockages and has not received any order from the police office, it asks for new directions to the latter and moves randomly while waiting for a response.

Additionally, when a police force has any information about stuck agents, it evaluates the distance to each of the targets and selects the closest one to act upon. We have used the Euclidean distance to calculate the distance between the police force and the targets.

Ambulance Team Agent Initially, ambulance teams search randomly for victims. When a victim is found, an ambulance team moves to that location and starts the rescue procedure. First, it analyzes if the building is on fire; in this case, unfortunately the ambulance team can do nothing, since it must avoid to die under the fire as well. If the building is not on fire, then the ambulance team tries to perceive whether there is a buried civilian, and starts the unburying process if necessary.

After the victim has been unburied, the ambulance team loads the victim and transports it to the closest refuge that it can find a path to. Once they arrive at the refuge, the civilian is unloaded.

At this moment, if the ambulance team does not have any other target in its memory, it starts to search randomly for another victim. Like in the other platoon agent strategies, if no victim is found within a short period of time, then it requests information to the ambulance center.

4 Experiment Results

We have tested the proposed *Partial Global* approach with the latest working simulator package *simulator-package.tgz* available at `http://sourceforge.net/projects/roborescue/`. We have run our test in two maps that are provided together with the Robocup Rescue simulator, namely Paris and Berlin. In order to run the test, we considered three kinds of platoon agents (Fire Brigade, Police Force, and Ambulance Team) as well as all corresponding tasks (extinguishing fire, unblocking streets, and rescuing civilians). Moreover, it was considered the existence of available center agents of each kind (Fire Station, Police Office and Ambulance Center).

The results presented in this section were obtained by using the proposed strategy implemented in Java code since we have not finished to rewrite our agents in *AgentSpeak* programming language to run over *Jason*. Nonetheless, at the competition we will have the complete proposed approach using the BDI architecture.

Table 1. Maps configuration parameters

	Paris	Berlim
Civilian	108	100
Ambulance Team	9	11
Fire Brigade	18	19
Police Force	20	9
Refuge	5	6
Fire Spot	5	3

We performed 10 simulations for each map, with 300 cycles each. Each simulation was composed of a number of agents like specified in Table 1. All the tests were run in a PC with processor Intel Quad-Core (2 GHz), 4 GB RAM.

The mean score final values obtained from Paris and Berlin maps are presented in Table 2.

We have faced many problems during the development process, which has resulted in the fact that not all the intended agents' features were working prop-

Table 2. Tests results

Map	Final Score
Berlin	27.28 ± 0.18
Paris	21.34 ± 0.14

erly. This condition is reflected in the results obtained on the Paris and Berlin maps, similar to the sample agents provided with the simulator.

In the sample runs, about 75 percent of the civilians were alive at the end of the simulations. The total health proportion of the civilians at the end of the simulation is nearly 0.7. However, fire brigades did not perform as expected and we intend to better investigate this fact in detail.

5 Conclusions

This paper presented a task allocation approach, named *Partial Global* approach, which is based on *Partial Global Planning* [4]. It considers and handles both local and global information in order to coordinate the agents in the Robocup Rescue Agent Simulation. While the local information is maintained by platoon agents, center agents maintain a global consolidated view of the environment, that was provided by platoon agents. Moreover, we propose the adoption of a BDI architecture and a multiagent programming language to implement the proposed strategy, instead of using pure Java.

The main problem found during the development was related to the lack of a proper documentation about the simulator's features. It delayed our understanding and coding procedure, since we needed to search for examples in source codes in order to understand what capabilities the simulator has and how it behaves. Moreover, differently from the previous simulator, it is not possible to select which messages we want an agent to hear, making communication control even harder; additionally, we believe that the simulator is far too heavy to run reasonably in a single computer. Consequently, we spent too much time just to test the agents' strategies.

We will continue to improve our strategy as long as we gain more knowledge about the simulator for this current and future competitions. We also believe that we are going to be able to improve the agents' strategy performance quite a lot in the next months for this edition of Robocup Rescue competition.

As future work, we intend to integrate an organizational model for multiagent systems based on notions like roles, groups, and missions in order to coordinate the tasks performed by each agent. A first idea is to use the MOISE+ model [5]. Moreover, we also intend to implement rescue agents that perform their tasks in a totally decentralized way, thus not requiring the existence of center agents in the environment for the propagation of information. Finally, we intend to implement the D* Lite algorithm [6] to perform the path planning routine.

Acknowledgements

André H. Pereira is supported by PIBIC/CNPq/USP. Jaime S. Sichman is partially supported by CNPq/Brazil.

References

1. R. H. Bordini, J. F. Hübner, and M. J. Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason.* John Wiley & Sons Ltd, Chichester, UK, 2007.
2. M. E. Bratman. *Intention, Plans, and Practical Reasoning.* Harvard University Press, Cambridge, USA, 1987.
3. E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
4. E. H. Durfee. *Coordination of Distributed Problem Solvers.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1988.
5. J. F. Hübner, J. S. Sichman, and O. Boissier. A model for the structural, functional and deontic specification of organizations in multiagent systems. In G. Bittencourt and G. Ramalho, editors, *Advances in AI*, volume 2507 of *Lecture Notes in Artificial Intelligence*, pages 118–128, Berlin, DE, 2002. Springer-Verlag.
6. S. Koenig and M. Likhachev. Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, 21(3):354–363, 2005.
7. A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. V. de Velde and P. J., editors, *Proceedings of the Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, volume 1038 of *Lecture Notes in Artificial Intelligence*, pages 42–55, London, UK, 1996. Springer-Verlag.
8. A. S. Rao and M. P. Georgeff. Modeling rational agents with a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, San Mateo, USA, 1991. Morgan Kaufmann Publishers.
9. A. S. Rao and M. P. Georgeff. Rational agents: From theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, USA, 1995.
10. M. Wooldridge. *An Introduction to Multiagent Systems.* John Wiley & Sons Ltd., 2009.