Experiments with the RoboCup Rescue Simulator in a post Earthquake emergency Italian Scenario

Angelo Biagetti
Dipartimento dei Vigili del Fuoco,
del Soccordo Pubblico e della Difesa Civile
Ministero dell'Interno

Alessandro Farinelli, Luca Iocchi, Daniele Nardi, Fabio Patrizi
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 ROMA, Italy
`[farinelli|iocchi|nardi|patrizi]@dis.uniroma1.it`

Abstract

In this paper we present the achievements of a research project, based on the RoboCup Rescue simulator, carried out in Italy in collaboration with the Italian Fire Department. The overall goal is to devise tools to allow for monitoring and supporting decisions which are needed in a real-time rescue operation in a post earthquake emergency scenario. As for the experiments, we have addressed the problem of task allocation by providing an experimental analysis of different strategies in different operative conditions. Moreover, we are currently extending the experimental analysis to feature-level information acquisition and integration. Finally, we discuss the limitation and potential for application of simulation based tools and, the RoboCup Rescue Simulation, in particular, based on the experience gained through the collaboration with the Italian Fire Department.

Keywords: *Resource coordination and information management in disaster scenarios, fire fighting and collapsed buildings (search and rescue, chemicals), distributed intelligence.*

1 Introduction

The RoboCup-Rescue Project started in 1999 with the goal of developing a comprehensive urban disaster simulator (see [8]). It aims at producing a software environment useful for both testing intervention strategies in a virtual world and supporting decisions in case of real disasters such as earthquakes or big fires.

The presentation addresses two issues: (1) extensions and supporting tools for the deployment of the RoboCup Rescue Simulator, (2) experiments and evaluation methodologies based on the simulations. With respect to the first issue we have developed some useful tools, that are shared with the RoboCup Rescue community in order to define new scenario and visualize agents' views as well as to design agents with information fusion, planning and coordination capabilities. In particular, we have modeled the Italian city of Foligno based on real data from the earthquake of the fall 1997 [4]. With regard to the second issue, we describe a methodology for evaluation of multi-agent system in this scenario that takes into account not only the efficiency of a system, but also its robustness when operation conditions and environment change, as well as other features, such as the ability to acquire a precise and coherent representation of the disaster scenario.

2 The RoboCup-Rescue Simulator: an application to the earthquake of Umbria and Marche

Below we sketch the overall structure of the simulator to provide some indications on the components that need to be developed in order to apply the simulator to a specific disaster scenario. For a detailed description of the simulator see [11].

The RoboCup-Rescue Simulator has a distributed architecture, formed by several modules, each of them being a separate process running in a workstation on a network. The following are the main components of the simulator: i) *Geographic Information System* - The GIS module holds the state of the simulated world. Before simulation begins, it is initialized by the user in order to reflect the state of the simulated area at a given time, then it is automatically updated at each simulation cycle by the kernel module. ii)*Kernel* - This module is connected to any other module. At each step it collects the action requests of the agents and the output of the simulators, merging them in a consistent way. Then the kernel updates the static objects in the GIS and sends the world update to all the connected modules. iii)*Simulators* - Fire-simulator, Collapse-simulator, Traffic-simulator, etc. are modules connected to the Kernel, each one simulating a particular disaster feature (fire, collapses, traffic, etc.). At the beginning of every simulation cycle, they receive from the kernel the state of the world, then they send back to the kernel the pool of GIS objects modified by the simulated feature (for example, a pool of burned or collapsed buildings, obstructed roads, etc.) iv)*Agents* - Agent modules are connected to the kernel and represent "intelligent" entities in the real world, such as civilians, police agents, fire agents, etc. They can do some basic actions, such as extinguishing a fire, freeing obstructions from roads, talking with other agents, etc. Agents can also represent non-human entities: for example they can simulate a police-office, a fire station, an ambulance-center, etc. v) *Viewers* - their task is to get the state of the world, communicating with the Kernel module, and graphically displaying it, allowing the user to easily follow the simulation progress.

In order to use the RoboCup-Rescue simulator in the context of the present project several issues must be taken into account.

The first issue we have addressed is the identification of the domain that should be based both on the availability of data and on suitability of the RoboCup Rescue simulators in modeling such an area. With respect to the simulation scenario a GIS editor for building the scenario for the RoboCup Rescue simulator has been implemented and data about the city of Foligno and the earthquake of Umbria and Marche (1997) have been acquired.

The second issue is the modeling of the agent system, which involves the communication infrastructure that allows the agents to exchange information during the simulation and the agent behaviour. An extension to the original RoboCup Rescue simulator that has been considered fundamental for our aims has been implemented in order to allow civilians to communicate with the coordinating agents of the Rescue forces. In addition, based on the models built in collaboration with the VVF, a basic set of agents have been designed; in particular, a prototype demonstrator, including firemen, police force and ambulances has been set-up to simulate the rescue scenario after earthquake. This has been achieved by providing a Cognitive Agent Development Kit (CADK, [2]), which allows for the specification of the behaviour of agents in a declarative fashion.

Finally, some issues concerning the viewers have been addressed. In particular, we have provided the system with the capability of visualising the specific views of individual agents to enable for evaluating the ability to reconstruct the situation after the disaster. Moreover, we have addressed the construction of a 3D model for providing a 3-D view of the simulation.

3 Evaluation Methodology

Evaluation of Multi-Agent Systems in the RoboCup Rescue domain is important not only within the RoboCup Rescue simulation competitions, but also for evaluating actual plans to be used during rescue emergencies.

Evaluation of MAS in the RoboCup Rescue domain is currently carried out within international contests [1, 7], by rating each competing rescue team with a score representing the result of its activity in a simulated scenario.

In the real world, however, events not always develop in a known and predictable way, since unexpected changes in the operative conditions and failures can occur at every time. The evaluation rule used in the RoboCup Rescue simulation competitions is applied under standard fixed conditions (only a particular known *configuration* is used) and thus it does not take into account the ability of a MAS to work under troublesome conditions and its ability to adapt to non-standard operative conditions (different *configurations*).

Below we summarize an evaluation method (see [3]), based on [6], which allows the analysis of a rescue team in a more realistic way, by analyzing the performance of a MAS in terms of efficiency under normal conditions, as well as in terms of reliability under changing working conditions.

3.1 Experimental settings

To acquire a measure of the reliability and the robustness of a MAS, a series of simulations are to be accomplished under changing operative conditions. These tests give a measure of the system adaptability to unexpected situations.

The operative parameters that can change during a simulation are different and we have grouped them in three classes, denoting the kind of agent capabilities that are affected: 1) Perception, 2) Action Execution, 3) Cooperation.

For each of these classes, we have selected a specific parameter that has been used for generating different operative *configurations*, and in this way we have characterized the following three tests, that have been performed under different configurations: i) **visibility test** that is performed by executing simulations to probe the activity of a system under different visibility conditions; ii) **disabled agents test** in which we have modeled those situations where agents may become suddenly not operational, making possible to analize the reactions of the MAS against new configurations of each force; iii) **noisy communication test** that introduces errors in the communication channel.

3.2 Performance measures

The performance of a rescue Multi-Agent System is measured in terms of efficiency and reliability. The efficiency is directly evaluated by the formula (also used in RoboCup-Rescue tournaments 2003):

$$V = (P + S/S_0) * \sqrt{B/B_0}$$

where P is the number of living agents, S is the remaining hit points (health level) of all agents, S_0 is the total hit points of all agents at initial, B is the area of houses that are not burnt and B_0 is the total area of houses at the beginning of the experiment; the higher the value of V for a rescue system, the better the results of the rescue operation.

The reliability, that is not considered in RoboCup Rescue Competitions, describes how much system efficiency is affected by the variation of *configurations*, and how much it depends on the values V assumed in the simulation sequence of a single test. To compute reliability we perform different simulations for the same *scenario*, changing a given set of parameters in the configuration worsening the abilities of the agents. Reliability is evaluated with the linear regression slope formula:

$$LRS = \frac{\sum_{i=0}^{N-1}(x_i - x_m) * (y_i - y_m)}{\sum_{i=0}^{N-1}(x_i - x_m)^2}$$

where (x_i, y_i) are the coordinates of a point in a Cartesian system, (x_m, y_m) the average values of these coordinates, N the number of points considered. To acquire the reliability value, this formula can be simplified with $x_i = i$ and $y_i = V(i)$, since each point of the graph represents the value of V obtained in the i-th *configuration*. Usually, the result is a negative value, since the effectiveness of the agents decreases with more difficult operative conditions. A small absolute value means a good degree of reliability of the system to adverse situations.

Obviously, this process can be applied to relevant measures other than the value of V. Depending upon the aspects of the system (i.e. task allocation or information fusion) to be analyzed, one can focus on different parameters, such as the number of messages exchanged among agents or the time during which agents are allocated, etc.

3.3 Performance comparison

Measures of efficiency and reliability of a single Multi-Agent System are of little significance if not compared with the results obtained from simulations of other rescue systems. Performance comparisons allow to establish the effectiveness of a new technique over the previous ones, or over the state-of-the-art.

Here we recall an example of the performance evaluation executed on four different rescue-systems (see [4] for a detailed description), created with the CADK tool (see also [2] for detatils) and differing for the information integration and resource allocation techniques employed.

To compare the performance of these four rescue systems, a set of experiments has been performed. The results are summarized by the diagrams in Figure 1.

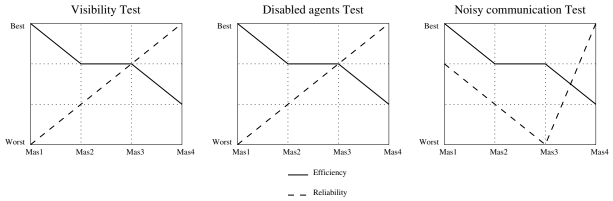

Figure 1: Performance comparison

In each test there is a rescue system which gets the highest value of efficiency and another one which obtains the highest score in reliability. Often in these tests the same rescue system is not the best for the two measures, since usually sophisticated techniques that improve efficiency turn out to be less robust to non-standard operative conditions, as the graphical representation of Figure 1 suggests. It is not obvious to identify which system has the best overall performance. In the visibility test, MAS1 is the best system in terms of efficiency, but it gets the worst rating about reliability. MAS2 and MAS3 have the same efficiency value, and are jointly ranked in the second place. MAS4, which is the worst system in terms of efficiency, is the best one with respect to reliability. The diagram shows also that MAS3 may be regarded as the best compromise between efficiency and reliability, since it is second in both of the two measures. In the noisy communication test, MAS1, which has the best efficiency value, is also a good system in terms of reliability, ranking in the second place; in this case, it seems to be superior to the other ones.

This example shows that the choice of the best system is hard to cast in absolute terms. Depending on the application, the system which offers the best score with respect to efficiency, reliability, or a (weighted) combination of the two may be selected. Indeed, the choice of a measure to select the best solution is a non-trivial task.

4 Task Assignment

In this section we show the performance analysis of a team of fire brigades with different task assignment strategies. The problem of task assignment in our simulation domain consists in the selection of agents to solve the three typical issues of the domain: (i) save civilians, (ii) extinguish fires, (iii) clean roads.

In order to study the task assignment problem in our target domain, we have focussed our attention on the issue of extinguishing fires by the fire force. The scenario is constituted by a city map in which fire agents and intervention places can be allocated on different regions of the map following different distribution. As we will see, different maps are suitable for different task allocation strategies.

In the first allocation strategy (named *Greedy-local perception* or GLP) no communication among agents is held and each agent goes to the nearest fire it can perceive. Notice that, given the strict communication limitation enforced in the RoboCup rescue competitions, several top performing team use similar approaches.

The second strategy is based upon a distributed mechanism, that regulates access to roles through the use of tokens. Each agent, when a new fire is perceived, creates a token referring to that role and decides whether to execute that role or pass the token to one of the team mates (using a round robin policy); each fire perceived through vision is registered in a list of known fires, hence only new fires are considered. This is useful to avoid too many agents going toward the same fire. Among the fires an agents knows, only the one at which the agent is most capable at is taken, the others are sent out. The capability of each agent is computed considering the distance from the fire. We referred to this strategy as *Token Passing* or TP. In both the strategies it could happen that two agents are competing for the same fire. However for the RoboCup rescue scenario this is not a problem, since agents help each other in fighting the same fire. Problems can arise if too many agents fight the same fire, because agents can block each other in narrow passages.

We have considered two scenarios: each one includes 18 fires and 10 agents, but in the first one fires and fire brigades are spread almost uniformly all across the city map (see Fig. 2), while in the second one fires are concentrated in one region of the map and fire brigades start from two regions of the map (see Fig. 3).

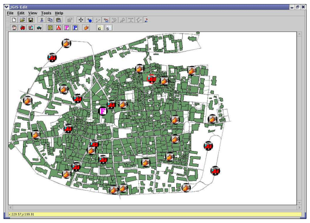

Figure 2: Scenario 1– Uniformly distributed agents and fires

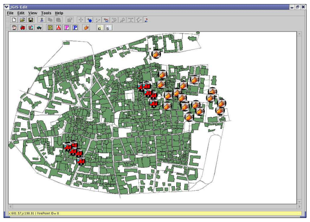

Figure 3: Scenario 2– Fires and agents concentrated in few regions of the map

From the results of each simulation, we have extracted the percentage of saved buildings, the extinguish time (time steps needed to put out all the fires), and finally the messages exchanged among agents.

	TP Sc. 1	GLP Sc. 1	TP Sc. 2	GLP Sc. 2
safe build.%	95.97	96.02	95.63	93.7
Ext. time	40.07	40	32.35	59
Msgs	359	0	811.35	0

Table 1: Task assignment results

Comparing the GLP allocation and the TP one we see that while the performance are similar in terms of saved buildings percentage and extinguish time, the GLP allocation does not need any message exchange, therefore it can be considered a better choice to the TP for such situations. The main reason to explain this result is that communication in the RoboCup Rescue simulator requires a considerably high amount of time: in fact it is possible to send at most one message each time step, which resemble approximately one message per minute in the real world, therefore when fires and fire brigades are nicely spread all over the city map, GLP allocation results actually as a very good strategy.

In the second scenario, where fires are concentrated in a particular region of the map, and fire brigades are concentrated in two other regions, communication is obviously more important and actually the TP approach consistently outperforms the GLP, while keeping a very low communication overhead. Notice that the scenario reported in figure 3, is much more likely to happen in a real rescue situation because, generally fire brigades starts their missions from specific fire centers while fires are mostly concentrated in few regions of the city map.

Summarizing, the experiments performed clearly show that the evaluation of a specific allocation strategy in the rescue domain should consider not only the efficiency and reliability of the strategy, but also particular features of the environment in which the allocation should work. As an example, in our particular reference scenario, the distribution of fires and fire brigades across the city map, is a fundamental characteristic to consider in the choice of the allocation strategy. In particular, if fires and fire brigades are spread on the city map in a uniform fashion, with a simple policy like the GLP it is possible to obtain good performance while avoiding messages exchange. However, for different distributions such as the one represented in Figure 3 more complex strategies are needed.

5 Information Fusion

In order to reach good performance in post-earthquake disaster situation agents need to exhibit both planning and cooperation capabilities, since the abilities of a single individual agent are often not enough for fighting an expanding disaster. However, another aspect to be considered, while developing a team of rescue agents is the need of integrating partial and noisy information coming from the agents, in order to assess a global situation, on which to perform the resource allocation. Indeed, the system performance is deeply influenced by the knowledge of the scenario. The interactions between information acquisition and integration and the process of decision making is clearly addressed in various models that have been proposed for information fusion [5, 10]. Specifically, at the stage of situation assessment not only is relevant the knowledge about the scenario, but also the effects and performance of actions taken in the scenario. It is worth noticing that, since agents may take specific actions to gather information (differently from passive sensors), the scenario is significantly different from that of a distributed sensor agent network [9, 13].

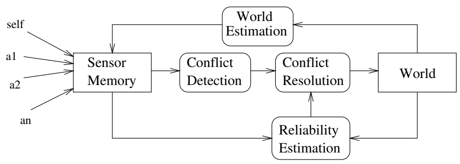

Figure 4: Information Integrator Module

The goal is to reconstruct the new state of the world starting from the previous states and the incoming information.

There are different levels at which the integration can be performed [12], depending on the properties of the data being integrated:
-*sensor-level data*: numeric values directly extracted from sensors;
-*feature-level data*: aggregation of numeric data representing specific features;
-*symbol-level data*: high level items related to agent knowledge.
In our approach the information integration is performed at symbol and feature level, in terms of properties of the world objects, since the RoboCup Rescue domain, in which we tested the system, is well suited for this level of information fusion.

In Figure 4 the functional structure of the Information Integration module of our CADK is sketched. The information provided by the sources, that can be on-board agent sensors, external sources (i.e. messages about the world situation coming from other agents) as well as agent expectations about the current situation are collected in a data structure (Sensor Memory). Then the possible conflicts arising from the comparison of the collected reports are detected and solved, by taking into account the reliability of the sources. Also the reliability is evaluated, by taking into account the evolution of the reconstructed situation. By choosing the level of information being integrated, as well as the conflict resolution policies, it is possible to implement a wide set of information integration strategies.

Information Fusion technology has the primary goal of improving the cognitive skills of a system, by integrating and correlating elements of different kinds. Roughly speaking, this amounts to obtaining from the raw data directly gathered from sensors compact, coherent and understandable information.

According to this view, an evaluation methodology is required, to measure the effectiveness of the fusion process, which addresses at least the following aspects: (i) quality of information, measured with respect to reality; (ii) robustness to noise and communication errors; (iii)computational effort.

To evaluate an information fusion system at data and object level, several techniques, based on the variation of some parameters and on the comparison of the obtained results [5] are used. However, the problem of evaluating the performance of a system for situation assessment and higher fusion level has received less attention, while we believe it is a very relevant issue in this framework.

6 Conclusions and future work

We have presented a methodology for evaluating the performance of a Multi-Agent Systems in the framework of a simulation environment for post earthquake scenarios. The aim of the work is to address the problem of devising methods and evaluating the performance of specific aspects of MAS. In particular, we reported some results on task assignment and presented a proposal for extending the approach to high level information fusion. Our idea is to attempt first to address the two aspects separately, in order to refine the experimental methodology and understand the specific features of each of them. However, in practice they coexist and deeply influence each other, since the management and coordination of a Multi-Agent System requires good knowledge of the situation where the operations take place. Therefore, after completion of the first phase, we plan to combine them and extend the approach to deal with both of them at the same time. In this extended framework the critical question of action vs perception, will be addressed in the context of a simulated MAS.

In addition, we are developing this work in collaboration with the Italian Fire Department, which provides us with technical expertise on rescue operations that we take into account in the design of both the simulation system and the MAS. In this respect, we are planning to use the tools developed in this project to build a scenario embodying features that are substantially different from those currently available, and to compare implemented strategies with those actually deployed by the forces of Italian Fire Department to make our working hypotheses in the simulation more accurate with respect to actual emergency scenarios.

References

[1] Robocup Rescue Web Site. http://robomec.cs.kobe-u.ac.jp/robocup-rescue.

[2] F. D'Agostino, A. Farinelli, G. Grisetti, L. Iocchi, and D. Nardi. Design and implementation of a Rescue-Agent Development Tool. In *Proceedings of the Workshop on Planning and real-time monitoring of rescue operations*, 2002.

[3] A. Farinelli, G. Grisetti, L. Iocchi, S. Lo Cascio, and D. Nardi. Design and evaluation of multi agent systems for rescue operations. In *Proc. of International Conference on Intelligent Robots and Systems (IROS'03)*, 2003.

[4] A. Farinelli, G. Grisetti, L. Iocchi, S. Lo Cascio, and D. Nardi. Using the robocup-rescue simulator in an italian earthquake scenario. In *Proc. of the 1st International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster*, 2003.

[5] D.L. Hall and J. Llinas (Eds.), editors. *Handbook of Multisensor Data Fusion.* CRC Press, 2001., 2001.

[6] G. A. Kaminka, I. Frank, K. Arai, and K. Tanaka-Ishii. Performance competitions as research infrastructure: Large scale comparative studies of multi-agent teams. *Journal of Autonomous Agents and Multi-Agent Systems*, 2002. To appear.

[7] H. Kitano and et al. Search and rescue in large scale disasters as a domain for autonomous agents resarch. *IEEE International Conference on System, man and Cybernetics (SMC99), Tokyo*, 1999.

[8] Hiroaki Kitano and Satoshi Tadokoro. RoboCup-Rescue: A grand challenge for multi-agent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.

[9] A. Knoll and J. Meinkoehn. Data fusion using large multi-agent networks: an analysis of network structure and performance. *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '94).*, pages 113 – 120, 1994.

[10] J. Salerno. Information fusion: a high-level architecture overview. In *IEEE Int. Conf. Information. Fusion*, AnnaPolis, July 2002.

[11] Robocup Rescue Web Site. http://robomec.cs.kobe-u.ac.jp/robocup-rescue.

[12] L. Valet, G. Mauris, and P. Bolon. A statistical overview of recent literature in information fusion. *IEEE Aerospace and Electronics Systems Magazine*, 16(3):7 – 14, 2001.

[13] Zili Zhang and Chengqi Zhang. Result fusion in multi-agent systems based on owa operator. *Proceedings of Australasian Computer Science Conference, (ACSC2000)*, pages 234 – 240, 1999.