Design and Evaluation of Multi Agent Systems for Rescue Operations

A. Farinelli, G. Grisetti, L. Iocchi, S. Lo Cascio, D. Nardi

Dipartimento di Informatica e Sistemistica

University of Rome "La Sapienza"

Via Salaria 113, 00198 Roma, Italy

E-mail : <lastname>@dis.uniroma1.it

Abstract— **The activities of search and rescue of victims in large-scale disasters are very relevant social problems, and from a scientific viewpoint raise many different technical problems in the fields of artificial intelligence, robotics and multi agent systems. In this paper we describe the development of a multi agent system based on the RoboCup Rescue simulator to allow monitoring and decision support, that are needed in a rescue operation. Two significant accomplishments are reported in this paper: the first is a framework for Cognitive Agent Development that provides for the capabilities of information fusion, planning and coordination; the second one is a methodology for evaluation of multi-agent systems in this scenario that aims at measuring not only the efficiency of a system, but also its robustness when conditions in the environment change.**

I. Introduction

Search and rescue of victims in large-scale disasters is a very relevant social problem, that poses several challenges from a scientific standpoint. When earthquakes, eruptions or floods happen, a considerable organizational capability to aid as fast as possible the disaster victims is required. This task is rather difficult since often different secondary disasters (e.g. fires, damages in the transportation and communication systems) connected with the main one, occur, which make the correct execution of a rescue plan a priori decided impossible.

In the recent past significant research initiatives have been undertaken in Japan [11], in the USA [7], and also in Italy [9], that specifically focus on the problem of developing software tools to support the management of this kind of emergency and, more specifically, to design a support system for search and rescue operations in large-scale disasters. These tools are intended both for on-line support during the actual operations as well as for previsional analysis and training.

In this paper we present the design development and evaluation of MAS operating in rescue scenarios, based on the RoboCup Rescue simulator. The results presented here have been the outcome of a research project that has lead to the development of a prototype tool, based on the RoboCup Rescue simulator, to allow monitoring and supporting decisions. The project has been developed in collaboration with the Italian Fire Department and uses as a case-study event the Marche and Umbria earthquake in Fall 1997.

The RoboCup-Rescue simulator [11] considers simultaneously three activities: (1) modeling of events related, in a direct way or not, to the disaster; (2) acquiring and integrating data coming from different heterogeneous sources; (3) modeling/monitoring/planning the resources used in the intervention. These activities offer an environment for experimentation, which provides significant advancements compared to the existing applications related to this field.

In order to perform the simulation on the chosen scenario we developed the GIS Editor, that allows for the creation of map files in a format suitable to the RoboCup Rescue Simulator, and for setting simulation parameters, such as fire ignition points or earthquake magnitudo levels.

In order to adapt the RoboCup Rescue simulator to an Italian scenario and experimenting several rescue policies, two main activities have been performed: (i) the design and implementation of different agents that act in the simulated world, under different policies, performing rescue operations; (ii) the experimental evaluation and analysis of the system components.

The first issue has required the development of a team of agents that perform rescue operations in the simulated world. In order to reach good performances in the post-earthquake disaster situation agents must exhibit both planning and cooperation capabilities, since the abilities of a single individual agent are often not enough for fighting an expanding disaster. Another issue 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. For experimenting different combination of information integration, cooperation, allocation and planning techniques in a structured way, we have developed a tool, the Cognitive Agent Development Kit (see Section III), that defines a modular agent structure for addressing the above issues, and provides a set of basic methods and primitives for quickly prototyping rescue agents with the above capabilities.

The second issue has been addressed in three steps: first, different multi agent systems implementing different communication, coordination, fusion and allocation techniques have been developed on top of the CADK, then an

evaluation methodology has been designed for evaluating the performance of a running multi agent system, and finally several experiments have been made in order to validate the proposed tools. The problem of evaluating a multi agent system has been approached in [10], [4] where tools are presented for performing controlled simulations in simple domains, but such evaluation techniques are strictly related to the specific application, while in [6] is proposed an evaluation method that has been experimented on MAS operating in soccer domain. Our approach is based on [6] since it focuses on aspects like coordination or robustness which are of interest also in rescue domain.

The rest of the paper is organized as follows: in Section II a short introduction to the RoboCup rescue simulation framework is given, in Section III the Cognitive Agent Development Kit is presented, finally in Section IV the evaluation methodology which is applied to the developed MAS is discussed.

II. ROBOCUP RESCUE SIMULATOR

The aim of this section is to provide a short introduction to the Rescue simulator, in order to give the basis for understanding the results persented in the next sections. The RoboCup-Rescue [11] simulator is a distributed framework made up by of a chain of simulators that operate on a virtual scenario, and interact each other through a set of services made available by a communication core: the kernel. Moreover such a module merges the world changes made by the simulators in a global world representation, and defines a network interface for the agents to operate. The simulation is realized in the discrete time system framework, where the state evolution in a certain instant is calculated on the basis of the state in the preceding instant, and of the inputs (represented by the agents/action). The sequence of actions forming a simulation frame is the following: state representation, catastrophic event simulation, simulation results integration, inter-module communication, state information transmission. The communication among the modules is based on an unreliable network protocol, and packet loss as well as modules responce delays can change the simulation outcome, as shown in Section IV.

III. COGNITIVE AGENT DEVELOPMENT KIT

The Cognitive Agent Development Kit (CADK) presented in this section has been defined in order to allow users to design and implement agents acting in a dynamic environment for accomplishing complex tasks. The agents developed with this tool have the following characteristics: (i) they can act autonomously in the environment by selecting the actions to be performed according to the information acquired from the environment; (ii) they can communicate with each other and cooperate to achieve a common goal; (iii) they can exchange information about the environment in order to reconstruct a global situation by using appropriate information fusion techniques.

All the agents are realized with three fundamental components (see Figure 1): 1) *Plan Executor*, that is responsible for executing a plan (i.e. suitably execute elementary actions that the agent can perform) for accomplishing a given task, the actual plan to be executed depends on the information coming from the *Coordination Manager*, which is described below; 2) *Information Integrator*, that is in charge of fusing the information about the world coming from its own sensors and from communication by other agents; 3) *Coordination Manager*, that is responsible for analyzing the current world state and the other agents coordination information, and choosing the agent specific goal (and thus the corresponding plan) in order to achieve a global goal for the team; the coordination protocol is distributed and thus it is robust to network failures and allows the agents to act autonomously [3].

In this section we describe an agent architecture that is suitable for modeling agents with the above described capabilities and operating in the RoboCup Rescue environment. Micheal Bowling [1] designed the Rescue Agent Development Kit, a library for developing agents operating in the RoboCup Rescue scenario. This is a useful development tool for the realization of multi-agent systems for the RoboCup Rescue simulator, since it provides a high-level interface to the programming of rescue agents, by hiding the agent-simulator communication details to the user. Our work on the Cognitive Agent Development Kit represents an extension of the original Agent Development Kit which provides all the rescue agents with planning, cooperation and information fusion capabilities.

In Figure 1 we describe the functional model of the agent. Agents are supposed to be able to communicate with each other through a proper device. Each oval in the figure represents a process, and each box represents a data structure. Such an architecture has been designed according to specifications that are suitable also for robotic agents, and, in fact, it has been actually used for developing cognitive mobile robots [5].

In the following we first present the data structures, then we sketch the main function of each process.

a) World.: The *World* is the structure that represents the whole agent knowledge about the world, it is composed by the rescue world objects such as roads, agents, buildings, etc., plus some information about the agent state. Note that the information contained in the *World* represent an *epistemic state* of the agent, that characterizes the knowledge of the agent about the world. This epistemic state is different from the actual state of the environment, since it contains only a partially and usually imprecise view of the current situation.

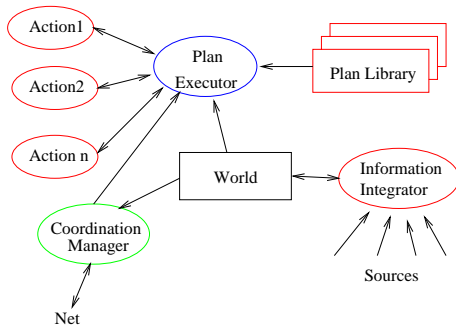*b) Plan Library.:* A plan may be seen as a program (in our case it is represented as a graph) that specifies

Fig. 1.   Functional Agent Description

Fig. 2.   Information Integrator Module

the actions an agent has to perform in order to reach a goal. Each node of such graphs is a state, while edges specify the state transitions caused by action execution. The plan library is a collection of plans. Each of those plan is indexed by its goal. These plans may be generated either by an automatic planner, [5], or by using a graphical tool.

c) Plan Executor.: The Plan Executor performs two tasks: 1) it receives from the Coordination Manager the goal to reach and peeks from the Plan Library the plan to achieve it; 2) it executes such a plan by activating and deactivating primitive actions at each state, and by considering the proper situation in case of a condition branch. A plan switch can occur either after the recognition of a plan failure or a when the goal is changed by the coordination module.

d) Information Integrator.: In general, an agent can be equipped with a large variety of sensors and/or sensing capabilities, each one providing a different type of input. The task of the sensor integrator 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 kind of data.

In Figure 2 the functional structure of the Information Integration module is sketched. The information provided by the sources, that can be on-board agent sensors, external sources (i.e. messages about the world situation
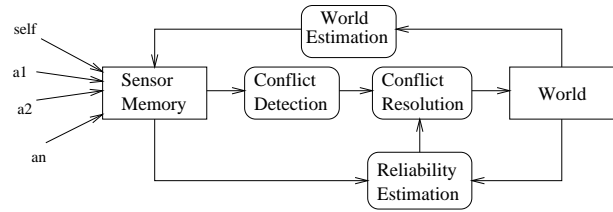
coming from other agents) as well as agent expectations about the current situation are collected in a data structure (Sensor Memory). Then the possible conflicts arising from the comparison of the collected reports are detected and solved, by taking into account the reliability of the sources. Also the reliability is evaluated, by taking into account the evolution of the assessed 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.

e) Coordination Manager.: There are many approaches that can be adopted in coordinating a multi agent system. It is possible to perform a fully distributed approach [2] as well as a centralized [13] one, or to combine both, leading on an hybrid coordination schema [8].

IV. EVALUATION OF MULTI-AGENT SYSTEMS

In order to evaluate the performance of rescue systems created with CADK, we present in this section a methodology based upon [6] which considers not only their efficiency under normal conditions, but also their reliability under nonstandard operative circumstances.

A. Experimental setting

To acquire a measure of the reliability and the robustness of a MAS, a series of simulations have been executed varying changing operative conditions. These tests give a measure of the system adaptability to unexpected situations. The parameters that we have considered for variation are: (i) perception radius; (ii) number of agents; (iii) errors in the communication system. Each parameter characterizes a particular series of simulations, referred to as the *visibility test*, the *disabled agents test* and the *noisy communications test* respectively.

1) The visibility test: In outdoor environments, visibility conditions are extremely variable. Rescue operations can occur every hour of the day, also in the night. Thus, it is necessary to probe the activity of a system also in these situations. The visibility test is performed by executing five simulations, each with decreasing perception radius, modeling activities under different visibility conditions (i.e. twilight, night time, fog). In this test, the varying conditions are on the perception range of each agent,

that is 30 meters under normal conditions, and we have performed experiments for the same multi-agent system also using 20, 10, 5 and 3 meters of perception range.

2) The disabled agents test: In a real emergency situation, it can happen that an agent suddenly becomes not operational for some unforeseen reason (for example, a mechanical failure of its vehicle or its equipment); this test analyzes the reactions of a system against new operative conditions, in which some of the operative agents are disabled.

The disabled agents test is composed of five simulations: in the normal conditions all the agents are active, for the other simulations one to four of the *best* agents for each force are disabled. The choice of the best agents to disable is based on the number of tasks performed: for each force, the agent that has completed more tasks in less time will be disabled.

3) The noisy communication test: Agent cooperation in the rescue domain is mainly attained by radio communications among coordination centers and between a coordination center and the operative agents. In real conditions, communication transmissions are not free from network failures, or human misunderstandings. This test verifies the robustness of an analyzed multi-agent system by introducing errors in the communication channel, thus preventing messages to reach their destination. The noisy communication test is composed of five phases: under normal conditions there are no errors in the communication channel, while in the other simulations 1/10, 1/3, 1/2, and 9/10 of the sent messages are lost.

B. 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 used in RoboCup-Rescue tournaments (2003), which is:

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

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** describes how much system efficiency is affected by the variation of operative conditions, and how much it depends on the values V assumed in the simulation sequence of a single test. 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 phase. 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.

C. 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 comparison allows to establish the effectiveness of a new technique over the previous ones, or over the state-of-the-art.

In this section, it is shown an example of the performance evaluation executed on four different rescue-systems, created with the CADK tool. The analyzed MAS differ for the information integration and resource allocation techniques employed, as shown in the following table:

Allocation\fusion	no fusion	simple
Static	MAS 1	MAS 2
Dynamic	MAS 3	MAS 4

To compare the performance of the these four rescue systems, the controlled experimentations depicted in the previous section are executed, giving the results shown in the left tables of Figure 3. In each test there is a rescue system which gets the best value about efficiency and another one which is best in reliability. Rarely in these tests the same rescue system is the best for the two measures, since usually sophisticated techniques that improve efficiency turn out to be less robust to nonstandard operative conditions. To provide better intuition to the previous results, a graphical representation is presented, sorting the results of each test with both Efficiency and Reliability, as shown in the right side diagrams of Figure 3. 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

Visibility Test	Effic.	Rel.
MAS1	23.3	−2.65
MAS2	18.8	−2.06
MAS3	18.8	−1.99
MAS4	11.6	−1.67

Disabled agents	Effic.	Rel.
MAS1	23.3	−6.18
MAS2	18.8	−5.32
MAS3	18.8	−5.09
MAS4	11.6	−4.88

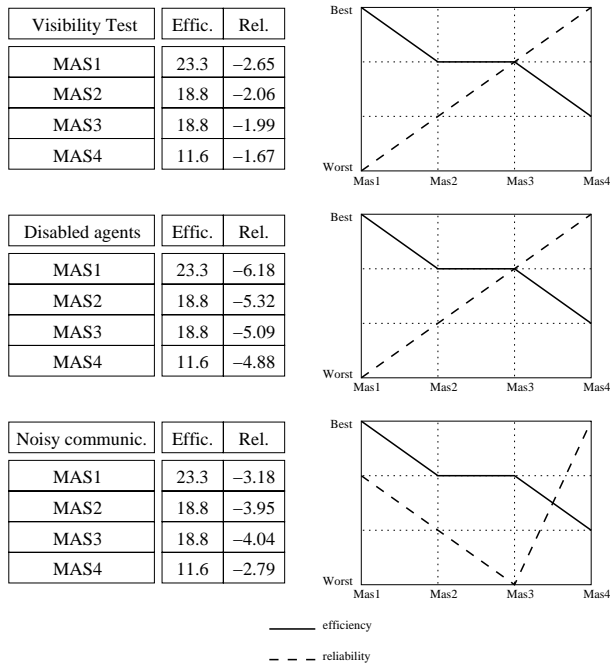| Noisy communic.	Effic.	Rel.
MAS1	23.3	−3.18
MAS2	18.8	−3.95
MAS3	18.8	−4.04
MAS4	11.6	−2.79

Fig. 3.   Performance comparison

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 to 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.

D. Result Variance in RoboCup-Rescue

Event evolution during a RoboCup-Rescue simulation depends not only on the initial setup and behaviour of rescue teams, but also on the system configuration where the simulation is running.

Since the simulation can be run along a distributed system, there are two main causes of unpredictable event development: (i) network reliability and (ii) processor speed. The first issue is crucial as the simulation system is composed by several modules (seven for the simulation and six for the agents), each of which communicates with the kernel module massively over the network at each simulation step. The protocol used by the modules to communicate is derived from UDP, thus causing some messages not to arrive at destination (ever or just in time). Processor speed is also crucial for simulation evolution since each agent has a limited time to plan an action; if the agent has not issued an action within this time it loses the turn. Hence, more sophisticated action planning techniques may cause the loss of action cycles, if not supported by adequate CPU power of the host machine.

For these issues the simulator system itself does not allow to replicate identical experiments. In this context, a statistical analysis of the results' variance of a robocup-resuce simulation can be useful for these two reasons: (i) it allows the analisys of rescue simulation system by giving a good estimation of the reliability of a particular hardware and software simulation enviroment, since low variance indicates few losses in network messages and agents' cycles; (ii) it gives an estimation of stability for a rescue team's behaviour: it is advisable to have low variance, since it means high robustness to unexpected event evolution.

As a sample analisys of RoboCup-Rescue variance we have executed 20 runs on the same scenario using the same initial conditions and the same rescue team; the results are summarized in the following table:

Name	Average	Deviation	Percentage
Evaluation	50.59	2.8	5.5 %
Saved civilians	23.75	3.5	14.7 %
Extinguished fires	16.45	6.2	37.7 %
Cleaned roads	313.6	28.0	8.9 %

In this case we can see that the analyzed rescue team is not so stable for the high variance of saved civilians, extinguished fires and cleaned roads; the worst parameter is the number of extinguished fires, which indicates poor performance of the fire force.

V. CONCLUSIONS

The aim of this research has been the design and evaluation of Multi Agent Systems operating in a rescue scenario, based on the RoboCup Rescue Simulator. The availability of the RoboCup simulator has been extremely valuable for the development of the project, providing an experimental setting that can be effectively used for developing a prototype implementation. We used as case study from the earthquake of Marche and Umbria in Fall 1997.

The main results of this work have been the development of a design methodology for MAS and of an evaluation criteria for MAS operating in a rescue scenario. As for the first issue we developed a new Cognitive Agent Development Kit, including functionalities like information fusion, planning and coordination needed for developing cognitive multi-agent systems, while for the second issue we have performed a set of systematic simulations in order to evaluate different rescue strategies. Moreover, this research has provided a significant use of agent technology in the design of tools supporting the acquisition of information as well as the planning of activities, when there is the need to act promptly and, therefore, with partial information about the situation, as in a typical emergency scenario. Finally, the evaluation of a multi-agent system based on the RoboCup Rescue simulator can be regarded as a step towards the evolution of actual plans to be used during rescue emergencies. To this end,

variance analysis gives important information about the simulation environment and the rescue team stability. We are planning to extend the work described here in two directions: (i) the development and the experimentation of new multi agent rescue systems on the top of the CADK framework, by designing new coordination methods and information fusion techniques (ii) the validation of real intervention strategies in collaboration with Italian Fire Department.

VI. REFERENCES

[1] Micheal Bowling. Robocup rescue: Agent developement kit. *available at official RoboCup Rescue Web Site,*, 2000.

[2] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa. Communication and coordination among heterogeneous mid-size players: ART99. In *Proc. of Fourth International Workshop on RoboCup*, pages 149–158, 2000.

[3] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa. Coordination among heterogenous robotic soccer players. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS'02)*, 2000.

[4] R.J. Firby and S. Hanks. A simulator for mobile robot planning. In *Proceedings of the DARPA Knowledge-Based Planning Workshop*, 1987.

[5] Luca Iocchi. *Design and Development of Cognitive Robots*. PhD thesis, Univ. "La Sapienza", Roma, Italy, On-line `ftp.dis.uniroma1.it/pub/iocchi/`, 1999.

[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] J. Llinas. Information fusion for natural and man-made disasters. In *Proc. of 5th International Conference on Information Fusion*, 2002.

[8] D. Miller. Multiple behavior-controlled microrobots for planetary surface missions. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 289–292, Los Angeles, USA, 1990.

[9] D. Nardi, A. Biagetti, G. Colombo, L. Iocchi, and R. Zaccaria. Real-time planning and monitoring for search and rescue operations in large-scale disasters. In *Technical Report University "La Sapienza" Rome*, 2002. `http://www.dis.uniroma1.it/~rescue/`.

[10] M. Pollack and M. Ringuette. Introducing the tile-world: Experimentally evaluating agent architectures. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 183–189, 1990.

[11] S. Tadokoro and et al. The robocup rescue project: a multiagent approach to the disaster mitigation problem. *IEEE International Conference on Robotics and Automation (ICRA00), San Francisco*, 2000.

[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] M. Veloso and P. Stone. Individual and collaborative behaviors in a team of homogeneous robotic soccer agents. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 309–316, 1998.