FC Portugal Rescue: High Level Coordination in the RoboCup Rescue Simulation

Nuno Lau, Luis Paulo Reis, João Certo

Resumo – **Este artigo apresenta os princípios da Liga de Busca e Salvamento (Rescue) do RoboCup e algumas das tecnologias desenvolvidas pela equipa FC Portugal desta liga. O RoboCup Rescue é um projecto conjunto internacional que promove a investigação em Inteligência Artificial Distribuída e Robótica Inteligente. Através desta liga é disponibilizado um simulador de desatres urbanos, fóruns de discussão técnica e avaliações competitivas das abordagens propostas pelos investigadores. Este artigo descreve os objectivos da liga, o modelo implementado pelo simulador, e os principais desafios de investigação que conduziram ao simulador do RoboCup Rescue. Neste simulador, agentes heterogéneos tentam minimizar os danos pessoais e materiais após a ocorrência de um terramoto numa cidade. Existem vários edifícios em chamas, civis feridos e estradas bloqueadas,. Equipas de bombeiros, polícias e ambulâncias colaboram para fazer face ao desatre. Neste contexto, é apresentada a equipa FC Portugal Rescue, incluindo o modelo estratégico de coordenação e os comportamentos principais dos vários tipos de agentes. São discutidos alguns desafios e as respectivas soluções da equipa FC Portugal. Alguns resultados obtidos em competições internacionais são também apresentados com destaque para o campeonato europeu de 2006 que foi ganho pela equipa FC Portugal.**

Abstract - **This paper presents an overview of the RoboCup Rescue Simulation League and the rescue team FC Portugal. RoboCup Rescue Simulation is an international joint project that promotes research on distributed artificial intelligence and intelligent robotics. It offers a comprehensive urban disaster simulator, forums for technical discussions and competitive evaluation for researchers and practitioners. This paper explains the objectives of the league, the mechanics of the simulator system and the main challenges in the research conducting using RoboCup Rescue simulator. In this urban disaster simulator, heterogeneous teams of agents try to minimize damage to both people and urban property after the occurrence of an earthquake. The city is filled with burning buildings, trapped civilians, and blocked roads. Teams of simulated fire brigades, policeman and ambulances collaborate in order to face the disaster. In this context, an overview of FC Portugal's rescue team is presented, including the strategic coordination layer of heterogeneous agents and implemented agent's roles and sub-tactics. In order to show the research conducted using the Rescue simulator some of the most interesting league problems are discussed together with FC Portugal's solutions. Results achieved in international competitions are also briefly presented, with emphasis on the 2006 European championship in Eindhoven, which FC Portugal won.**

I. INTRODUCTION

The Rescue Simulator is an environment for Information Systems (IS), Artificial Intelligence (AI), Multi-Agent Systems (MAS) and Intelligent Robotics (IR) research. The concept of Multi-Agent Systems evolved from Distributed Artificial Intelligence (DAI), Distributed Problem Solving (DPS) and Parallel AI (PAI). As for a single intelligent agent, it can be defined as a computational entity, usually called software, that if placed in some environment, perceives it through sensors, and is capable of performing autonomous action, in order to meet its design objectives, using its actuators [1]. A Multi-Agent System is an environment where several agents are inserted and where they can interact. RoboCup was created as an international research and education initiative, aiming to foster research in (distributed) artificial intelligence and (intelligent) robotics research, by providing standard problems, where a wide range of technologies can be examined and integrated. With the objective of dynamizing the evolution of AI, IR and in particular MAS, the project was launched by Hiroaki Kitano, a Japonese AI researcher that became founder and president of the RoboCup Federation. It is currently divided in four major categories: soccer, rescue, @home and junior; each with its different leagues. Due to its prominence, soccer was the main motivator behind RoboCup. Being an extremely popular sport across most of the globe, it is able to attract people from different countries, cultures and religions into the same competition. Furthermore, it presents interesting scientific challenges, mostly because it is a team game, mingling individual efforts with collective strategy.

RoboCup Rescue Simulation is an international joint project [2] that promotes research on distributed artificial intelligence and intelligent robotics in a scenario which is more useful to the society than soccer. The project started in 1999 to solve rescue problems, integrating disaster information, prediction, planning, and training for rescue actions. Built upon the success of RoboCup Soccer project, it aims to offer a comprehensive urban disaster simulator, forums of technical discussions and competitive evaluation for researchers and practitioners [3]. Through the use of an extensive, and ever evolving, urban disaster simulator, an heterogeneous team of agents tries to minimize damage to both people and property. Burning buildings, Civilians trapped under debris, and blocked roads, are just some of the challenges simulation Rescue Teams (RTs) must overcome, coordinating as many as up to forty agents of six different types [4].

Every year, a RoboCup international competition is organized, where, in a competitive but constructive environment, researchers from all over the world can test their agents against other RTs. By comparing approaches and exchanging ideas, progress is made at an amazing rate, in great part due to the open source nature of the project. After each competition, the source code for every team is released so that work may be done on top of the best ideas and implementations. Following on that concept, team FC Portugal entered the Rescue project, determined to contribute to the community. The agents' base work was built on top of the code developed by SOS [5], a reputed RT from Iran. At the top of the RoboCup Rescue agents, strategic and coordination methodologies from the RoboCup Soccer are being used. The constant evolutions in the simulator package imply that a lot of effort is required simply to adapt rescue agents to new environment rules. FC Portugal Rescue team is the result of a cooperation project between the Universities of Porto (LIACC/NIAD&R Lab) and Aveiro (IEETA Lab) in Portugal.

The rest of the paper is organized as follows. Section 2 presents RoboCup Rescue simulator, its main modules and the associated viewer system. Section 3 describes the process of construction of a RoboCup Rescue team and the main methodologies and algorithms needed for this task. Finally, section 4 presents some results and the conclusion of the paper.

II. THE SIMULATOR SYSTEM

The Rescue simulator is a simplified model of a city - only data relevant to the disaster situation is reproduced, neggleting most of the unrelated detail. The simulator package uses a modular approach, allowing different parts to be updated independently. Every year new features are combined with the existing ones, improving the simulation and adding complexity to the environment. The most recent large change was in the fire simulator, which was completely overhauled, requiring some changes in the agents' strategies.

The action takes place in a simulated city, where a natural disaster (earthquake) has just taken place. From the initial situation the following values can be obtained:

- Sint: total HP of all agents at start.
- Bint: total undamaged area at start.

At any time step those values can be obtained:

- P: number of living agents.
- S: remaining total HP of all agents.
- B: total undamaged area of buildings.

The simulation score V is calculated using the following equation:

$$V = \left(P + \frac{S}{Sint}\right) * \sqrt{\frac{B}{Bint}} \qquad (1)$$

From Eq. (1) it can be perceived that the evaluation rule would be: given any simulation, the higher the V value, the better the rescue operation [6].

Note that at the beginning of the simulation (t=0):

$$V = (P + 1) \qquad (2)$$

In Eq.(2), which results from Eq.(1) at t=0, the initial score is defined as the total amount of agents plus one. As the simulation proceeds, more buildings are damaged and people hurt, causing the score to drop till its final value at t=300.

A schematic representation of the simulation system can be seen on Figure 1.

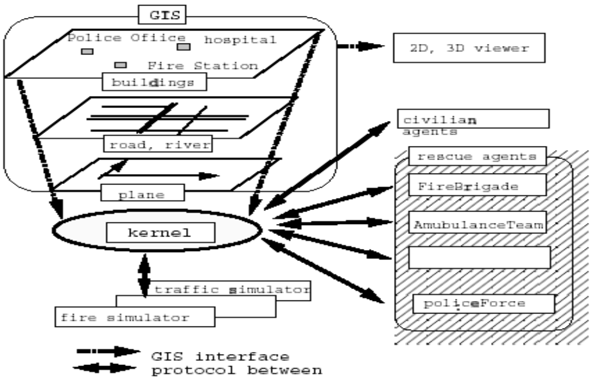

Figure 1 - Simulation System functional outline [7].

This structure allows a relatively autonomous development of the different simulator modules, since once the communication protocol is defined, the modules are mostly independent. The communication between modules takes place by message exchange.

The kernel is the central processing unit of the system, controlling the simulation process and facilitating information exchange between modules. It is responsible for establishing and maintaining communication with the Geographic Information System (GIS), the Simulators (Collapse, Fire, Traffic, etc.), the Viewer, and the Agents; as is depicted on Figure 2.

When the program starts, the Kernel receives from the GIS module the initial configuration of the simulated world. At every step of the simulation, the Kernel sends sensory information to all agents and receives their action commands. Information is sent and received from the modules as necessary and, for each data exchange, the command and information validity is verified [8].

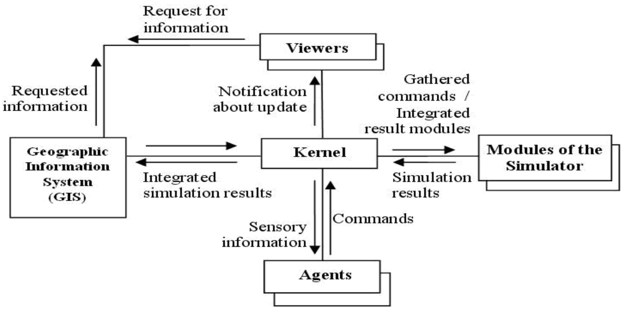

Figure 2 - RoboCup Rescue Simulation System.[8]

The GIS module is responsible for the initial configuration of the simulated world. This is composed by the location and properties of buildings, roads, nodes, refuges, agent centers, Civilians, Ambulances, Fire Brigades, Police Forces and initial fires. It also records the simulation progress into a log file, enabling a detailed offline analysis. Additionally, this module is responsible for feeding data to the viewer.

The Collapse Simulator module acts on the physical state of buildings after the earthquake. On a large scale disaster like the one RoboCup Rescue aims to emulate, around 80% to 90% of households are at least partially collapsed, shortly after the calamity. Currently, this simulator is triggered only once, at the beginning of the simulation.

The Blockade Simulator module is responsible for defining the state of road obstructions. After the earthquake, a large part of the roads gets blocked, hindering traffic flow. These obstructions may have different causes such as crowds, debris from buildings and traffic accidents. Blocked roads can only be cleared by Police Force agents, this way allowing other agents to freely move through.

Every agent's movement in the world, including Civilians, is modelled by the Traffic Simulator component, which defines the pace allowed on every road section. Width, number of agents present, and the level of "blockness", are some of the factors affecting maximum speed on a street. Usually, a road which is over 50% blocked is not traversable.

The Fire Simulator module simulates the spread of fire in the city. It is currently one of the most evolved components of the simulator package. Right after the earthquake, some buildings ignite and start radiating heat to nearby structures. This component is responsible for the physical simulation of combustion and heat spread. This is done resorting to an intelligent model in which the temperature of a building is, on the one hand increased, either by the its own combustion or by the radiation waves from neighbouring buildings, and on the other hand decreased, due to the evaporation of water, pumped by Fire Brigade agents. In the simplified combustion model used, the critical factors are temperature and fuel (buildings), with the supply of oxygen being disregarded. When a building's temperature rises above its material's flash point it bursts into flames, as seen on Figure 3. In contrast, when the temperature drops bellow this temperature, its fire is extinguished [9].

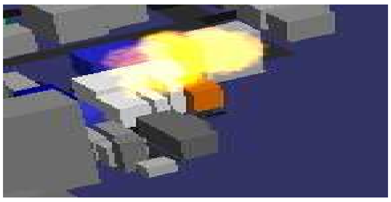

Figure 3 - An image from Freiburg's 3D viewer shows a building on fire.

The agent's status is modelled by the miscellaneous simulator. When an agent is inside a burning building or trapped under debris, its health is affected and starts decreasing. This is the module responsible for controlling the agent's properties in these situations. As a simple example, a large value for the agent's property buriedness describes its state as trapped under debris. When Ambulances use their rescue ability, this value is progressively reduced until the agent is free.

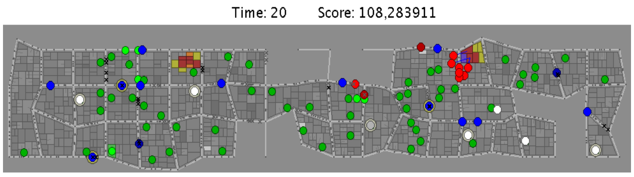

Figure 4 - Morimoto Viewer displaying simulation in RandomMedium map (RoboCup 2005)

A viewer is the graphical interface used to display the actions taking place in the simulated city. A snapshot of the Morimoto 2D Viewer can be found in Figure 4. This viewer shows agents as colored circles. Ambulance Teams are white; Fire Brigades red; Police Forces blue; Civilians green and all of them get darker when hurt, turning completely black if they die. Buildings also have different colors, according to their function or status. While refuges are green and Center (agent) buildings are white, those on fire evolve from yellow, to orange, to red. Flooded and extinguished buildings have different shades of blue, while those burnt down are dark grey (almost black). Roadblocks are marked with crosses, and current time and score are displayed on top of the map.

III. STRATEGIC LAYER MODEL

The strategic layer model provides a structured method of representing, building and managing a strategy in a scenario where a team of agents is used to perform a given cooperative task [10]. The terms scenario and agent should be considered as broader terms. Scenario can be a simulation, a game, a simplified view of the real world or any other kind of setup where there is an environment where a team of agents has one or more objectives to fulfil.

This model can be used to manage the strategy of a heterogeneous Rescue team, but is is designed to be general and to be applied in static, dynamic, reactive or nonreactive environments.

In order to better explain the model, a top-down approach will be followed.

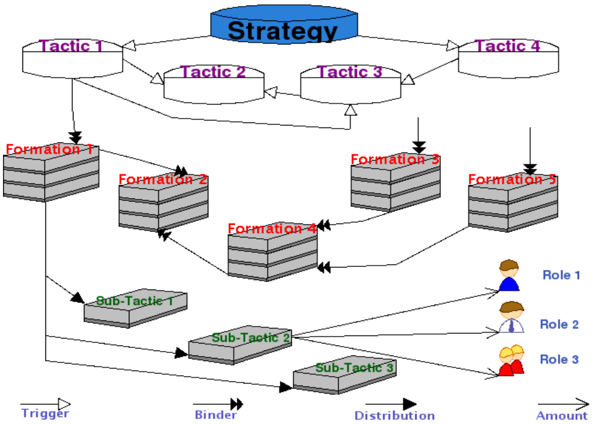

Figure 5. Schematic of strategic concepts.

Figure 5 represents the proposed model and depicts the interconnections between the concepts presented in this model. The figure only expands one branch for each concept.

A. Strategy.

Informally, a strategy is the combining and employment of means in large-scale, long-range planning and the act of directing operations for obtaining a specific goal or result.

Formally, a strategy is a combination of tactics used to face the scenario and the triggers to change between tactics:

$$\text{Strategy} = \{\text{Tactics}, \text{Triggers}\} \qquad (3)$$

A strategy can have several available tactics:

$$\text{Tactics} = \{\text{Tactic}_1, \text{Tactic}_2, \ldots, \text{Tactic}_t\}, \ \Box\ t \in \mathrm{N} \qquad (4)$$

Triggers set the conditions to interchange tactics:

$$\text{Triggers} = \{\text{Trigger}_1, \ldots, \text{Trigger}_{tg}\}, \ \Box\ tg \in \mathrm{N} \qquad (5)$$

B. Tactic.

A tactic is an approach to face the scenario in order to achieve a goal. Tactics deal with the identification of different situations and the correspondent use and deployment of agents in the scenario for those situations.

Formally, a tactic defines agents' formations as the arrangement of agents, situations as the combination of scenario conditions that can be seen as a more particular problems and binder as the association between a formation and a situation or between several situations and a formation.

A tactic should be self-sufficient, i.e., it does not need other tactics to function through all the simulation. There can be only one tactic active at one given time.

$$\text{Tactic} = \{\text{Formations}, \text{Situations}, \text{Binders}, [\text{Tactical Parameters}]\} \ (6)$$

A tactic has several formations that can be used:

$$\text{Formations} = \{\text{Formation}_1, \ldots, \text{Formation}_f\} \ \ \Box\ f \in \mathrm{N} \qquad (7)$$

A tactic also defines different, useful, situations:

$$\text{Situations} = \{\text{Situation}_1, \ldots, \text{Situation}_s\}, \ \Box\ s \in \mathrm{N} \qquad (8)$$

Tactics have binders in order to associate formations with situations:

$$\text{Binders} = \{\text{Binder}_1, \ldots, \text{Binder}_b\}, \ \Box\ b \in \mathrm{N} (9)$$

In a situation, the conditions that make it unique are defined:

$$\text{Situation} = \{\text{Condition}_1, \ldots, \text{Condition}_{cd}\}, \ \Box\ cd \in \mathrm{N} \qquad (10)$$

A binder sets the situations that lead to a formation. Optionally, a binder can set the connection between several origin formations and a terminus formation through situations:

$$\text{Binder}=\{[\text{OrigFormations}], \text{Situations}, \text{TermFormations}\}, \Box \ [\text{OrigFormations}], \text{TermFormation} \in \text{Formations} \quad (11)$$

C. Formation.

A formation is a high-level structure that aggregates all the agents with the intent of assigning them to specific sub-tactics. The aggregation is either wrought by using agents that belong to the same type, have the same more immediate goals, or both.

Formally, a formation is a specific association of sub-tactics with a defined distribution that may specify an agent type. Only one formation can be active at any given time. As such, the formation must include sub-tactics for all agents.

$$\text{Formation}=\{\text{Distribution}, \text{Sub-Tactics}, [\text{AgentTyp}]\} \ (12)$$

The same sub-tactic can be used more than once in a formation. This allows an implicit definition of Group. Let sub-tactics be a multiset where m(Sub-Tactic st) defines the multiplicity of a sub-tactic:

$$\text{Sub-Tactics}=\{(\text{Sub-Tactic}_1, m(\text{Sub-Tactic}_1)), \ldots, (\text{Sub-Tactic}_{st}, m(\text{Sub-Tactic}_{st}))\}, \ \Box\ st \in \mathrm{N} \ (13)$$

For each element in sub-tactics there is correspondent value in a distribution:

$$\text{Distribution}=\{\text{Value}_1, \ldots, \text{Value}_v\}, \quad v = \sum m(\textit{Sub-Tactics st}) \quad (14)$$

A distribution specifies either absolute or percentage distribution values for each sub-tactic in the formation. Distribution values always refer to agent types when applicable. In this manner, the total of values can surpass 100%, but not for a specific agent type.

The association with agent type is implicit when a sub-tactic can only be applied to one agent type. Otherwise, when more than one agent type can be used (see section III.E), an agent type must be specified for that sub-tactic:

$$[\text{Agent Types}] = \{\text{Type}_1, ..., \text{Type}_{ty}\} ,\ \square\ ty \in N \quad (15)$$

D. Sub-Tactic.

A sub-tactic reflects the approach to face the scenario of a limited set of agents either partially for a number of situations or during the whole scenario.

Formally, a sub-tactic is an association of roles with one default amount of agents assigned to those roles. Additionally a sub-tactic may also have sub-tactical parameters to reflect specific thresholds, agent parameters, coordination options or other values that are needed to configure the roles used on the sub-tactic.

$$\text{Sub-Tactic}=\{\text{Amounts, Roles, [Sub-TacticParams]}\} \quad (16)$$

A sub-tactic can have one or more roles:

$$\text{Roles} = \{\text{Role}_1, ... , \text{Role}_r\},\ \square\ r \in N \quad (17)$$

For each role in sub-tactic there is an amount in amounts:

$$\text{Amounts}=\{\ \text{Amount}_1, ... , \text{Amount}_a\},\ a=\sum role\ \text{r} \quad (18)$$

Like in a distribution, an amount specifies either absolute or percentage values for each role in the sub-tactic. Percentage amounts in a given sub-tactic must total 100%.

Sub-tactics can be divided into Typed Sub-Tactics and Generic Sub-Tactics. In a typed sub-tactic at least one of the roles is associated with an agent type, which becomes the sub-tactic's type.

In order to ease the handling of different agent types, it is not possible to use roles of different agent types in the same sub-tactic. As such, typed sub-tactic can only use roles for one agent type together with generic roles. As a consequence, to build a formation with different agent types, there should be at least one sub-tactic for each agent type.

A generic sub-tactic is a particular kind of sub-tactic without any association with an agent type. Thus, in a generic sub-tactic, only generic roles can be used. As it was previously stated, if a generic sub-tactic is used in a formation that contains sub-tactics for more than one agent type, an agent type must be specified. This type is specified together with a distribution value when agents are assigned to a generic sub-tactic.

In the event that there are no agent types, or there is only one type of agent in the tactic, all sub-tactic kinds are generic and can be refereed simply as sub-tactic.

E. Role.

A role is a normal or customary activity of an agent in a particular environment.

Formally, a role is a set of algorithms in a defined sequence that describes an agent's behavior. The behavior description is expected to include, when relevant, the specification on how the agent should coordinate with agents in the same role or in other roles.

The agent coordination can be of three different kinds:

- All agents with the same role form one group;
- All agents with the same role form several smaller groups (with a rule specified inside the role);
- All agents with the same role act individually.

The role also defines partial objectives accordingly to the coordination method used. Although roles can describe the behavior for an entire scenario, they can also describe the behavior for only a given time frame or situation. Teams form their roles by combining different motion and action mechanisms with partial objectives.

The role level is the lowest in the proposed model. For teams who use sequenced task/objective/state based agents, a conversion to role based agent is discussed in section III.G.

Similarly to the sub-tactics, roles can be divided into Typed Role or Generic Role. A typed role is a particular kind of role that can only be assumed by one agent type. Typed roles are use when, in heterogeneous agents, there is a need to use the different agent's properties or capabilities. A generic role is a kind of role that can be assumed by any of the agent types used in a tactic.

F. Decision, Supervising and Communication

The decision maker depends on the agents' organization and types set by the scenario. In teams where there is only a supervisor and all the agents are “dummy”, the strategical layer will obviously only be applied to the supervisor.

In multi-agent systems, the first rule is that all agents have full knowledge of the strategical layer being used. Then if all agents have a good, shared, world state knowledge using the layer can be done with no extra communication. This is accomplished because all the agents switch their tactics, situations and formations based on the same conditions and at almost the same time. When a team uses a mechanisms like ADVCOM [11], the no-communication version of the strategical layer can be applied to scenarios were the normal communications are limited and unreliable.

If agents have more limited computational resources but still have good world state knowledge synchronization, the layer can be computed only by a supervising agent. This supervising agent would only have to communicate a new formation whenever declared by the strategical layer.

The supervising agent is chosen taking into account the agent who normally has more computational resources. Some scenarios specifically have supervising agents. In environments where the world state sharing is unreliable, the layer must be computed by a supervising agent choosing typically, the best informed agent.

G. Example for the Rescue Team

In order to adopt the strategic layer, our rescue team needed to use role concept. The previous code was based on a sequential selection of algorithms based on world state conditions. To reach the role level the following classifications were used:

- Action: a simple deed performed by an agent. E.g.: Action: Refill; Description: filling a Fire Brigade tank in a refuge.
- Task: set of actions performed by an agent that leads to a goal. E.g.: Task: Rescue civilian; Actions: Move to civilian; Unbury Civilian; Load Civilian; Move to refuge; Unload Civilian.
- Algorithm: set of tasks performed by one or more agents used to solve a particular field problem in a specific manner. E.g.: Algorithm: Clear main roads by prioritizing the main roads; Tasks: Each chosen road or set of roads is assigned to a specific Police Force, and then Police force agents clear the roads.

After identifying the algorithms, they were associated into roles. Some partial, generic roles like finding civilians were also created. Although these roles only included algorithms related to search and dislocation and do not have algorithms to act after all civilians are found, they are extremely useful.

The following figures depict a simplified rescue strategy. In Figure 6 the strategy is only expanded in one tactic and one formation. As shown, there is a different initial tactic depending on city size, T1 for small cities as T3 for large.

For a large city (T3) the losses will be unavoidable so a tactic that marks city zones as lost from the start would be more effective.

For a small city (T1) an option to focus on human life was made so, at start, agents will be more focused on finding and rescuing civilians. When more than 60% of known civilians are rescued and 80% of the buildings are explored, the tactic changes to T2 giving priority to fire fighting.

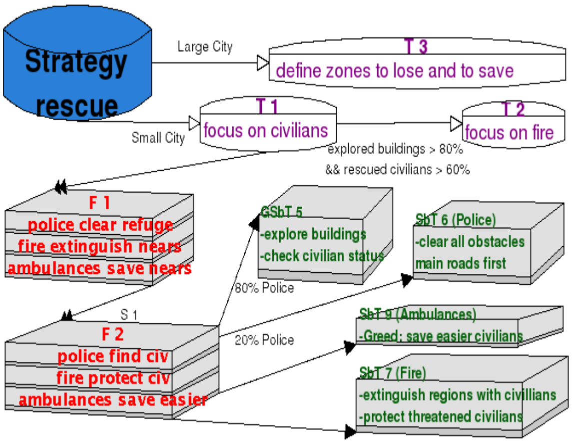

Figure 6. Partial rescue strategy.

Tactic 1 has two formations: the initial F1 and F2. F1 is used to ensure that rescue agents are saved as soon as possible and that the refuges are reachable. Note that refuges are essential buildings as Fire Brigades use them to refill their tanks and Ambulances Teams to unload civilians.

Formation 2 is used to find, protect and rescue civilians. Instead of focusing on unblocking roads, Police Forces explore buildings trying to find civilians. Likewise, Fire Brigades opt for extinguish buildings near trapped civilians instead of minimizing fire spread. In Figure 7 the situation (S1) to switch from formation F1 to formation F2 is defined.

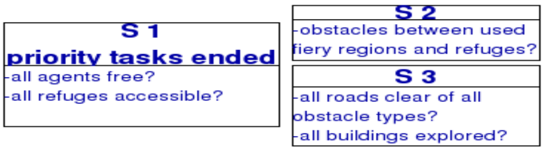

Figure 7. Some rescue situations.

In Figure 8 the sub-tactic SbT 7 is expanded. In this sub-tactic 80% of the Fire Brigades assume the role of protecting civilians that are directly threatened by fire. The remaining Fire Brigades chose to put out fires in city regions with civilians.

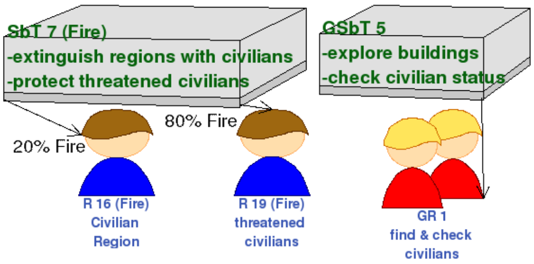

Figure 8. Two rescue sub-tactics.

The generic sub-tactic GSbT 5 has 80% of the Police Force assigned to it (Figure 6). In Figure 8 is seen that all of those agents are assigned to the generic-role Gr1 used for exploring the city in search of civilians and to checkup on their status. As Gr1 is a generic role it could also be assumed by Ambulance Teams or Fire Brigades yet, Police Forces can unblock roads in their path thus reaching any building which was found to be more useful in this case.

IV. MODELLING BEHAVIOURS

A RoboCup Rescue Team is composed of three kinds of field agents (Fire Brigades, Ambulance Teams and Police Forces) and their respective Center Agents. Field Agents have a limited eyesight and voice range of ten and thirty meters respectively. The simulated map area is usually in the order of a few dozens of square kilometers. In this section, the implementation of sub-tactics and roles of FC Portugal's agents is discussed.

A. Ambulance Teams and Ambulance Centers

The main function of an Ambulance Team is to unbury Civilians and take them to a refuge. The sub-tactics for Ambulance Teams follow.

Based on the known Civilian properties, mainly buriedness and health points, Ambulances estimate the time of death and schedule the order in which Civilians should be saved. The time taken in travelling, and whether a path to the Civilian exists, are considered.

The basic procedure for an Ambulance is the following: go to the Civilian position; unbury him; load him into the Ambulance; travel to the closest refuge and unload the Civilian to safety, moving on to the next one. However, this behaviour can change due to several events, such as receiving updated or new information about a Civilian, which may change rescue order and priorities, and possibly lead to a tactic change. Behavior also changes if there is a fire in a building with a Civilian. Buried field agents will always have higher priority.

In FC Portugal's implementation, Ambulance Teams can form and act as a group. This happens because the action of unburying an agent is cumulative and directly proportional to the number of Ambulance Teams. Therefore, the more Ambulance Teams, the faster the agent will be unburied. There are some exceptions to this tactic. Since the moving cost is considered when estimating the Civilian time of death, sometimes it is more efficient for Ambulances to act individually. Another exception is at the beginning of the simulation given that, due to road blocks, Ambulance Teams don't usually have free paths to the same agent, so they also act individually rescuing the civilians they can get to.

As Ambulance Teams' calculations are relatively simple, there is no need for further calculations at the Ambulance Center and, as such, its main function is to act as a repeater. It should be kept in mind that Ambulance Teams take several cycles to unbury an agent, which is more than enough time to compute the next to be saved.

B. Fire Brigades and Fire Station

Fire Brigade agents are the most complex field agents and their function is to extinguish fires. The sub-tactics for Fire Brigades are the following.

Depending on map size, Fire Brigades are organized into groups - usually two or three. Based on relative positions, size, and proximity to refuges, fiery regions are prioritized, and the ones with the highest priority are assigned, sequentially, to the available groups. Each group then considers its assigned fiery region, and prioritizes the burning buildings (from now on they are called targets) to extinguish, based on relative position in the fiery region, percentage of building area unburned, proximity with building with buried Civilian, amongst other factors.

The next step is to choose a suitable neighbor building that is in the water range of the target. The conditions for this are: the building cannot be on fire; it must be reachable; and as close to the target as possible. If no suitable building is found, then a road near the target is used. The disadvantage to this solution is that the Brigade occupies a lane, which in turn increases the risk of a traffic jam. After moving to the selected building, the Fire Brigade starts watering the target. Note that if for some of the above stated reasons the target changes, and if the new target is in range of the water cannon, the Fire Brigade starts watering the new target without using a move action.

After some cycles, water in tanks is depleted and, therefore, Fire Brigades go to the nearest refuge to refill their tanks. As this action takes some cycles, when refilling finishes, Fire Brigades reprioritize between the previous assigned region and the currently unassigned ones - the Fire Brigade proceedings are then repeated. Logically, if at some point there are more groups than fiery regions, the available group will be assigned to a fiery region using the remaining criteria.

Akin to Ambulance Teams, when Fire Brigades run out of fires to extinguish, the groups are scattered, so that new fiery buildings may be found. If none is discovered, Fire Brigades start searching for civilians. When all buildings have been explored Fire Brigades keep on visiting all known living and buried civilians keep information on their properties updated, allowing Ambulances to better estimate the civilian time of death.

Fire Station agent's main function is to act as a repeater; however, some of the calculations computed by Fire Brigade agents are also computed by the Fire Station. This happens both in the prioritize regions case, and in group assignment. As seen, cycle time is limited and, sometimes, field agents have to rush decisions. Fire Stations do not have this limitation, thus compare the field agent's solution to their own and, if the Station's solution is better, it is sent to Fire Brigades.

C. Police Forces and Police Office

Police Force agents' function is to clear blocked roads. The first strategic decision made is to only clear road blocks at not passable roads. This means that partially obstructed roads will not be cleared, and the reason for this is that they only affect the speed of an agent by halving it. The speed reduction isn't significant, when compared to the time it would take a Police Force to move to that road and remove the partial block.

The main problem for Police Forces has to do with the order in which road blocks are removed. On FC Portugal's implementation there are several ways to do this:

- Clearing blocks until a refuge is reached;
- Clearing blocks from a specific point to a refuge;
- Clear blocks around a refuge;
- Clear a specific path;
- Clear a specific cell.

These possible options are called tasks. Each of these tasks is given a weight, which is a parameter specified in a Police Force configuration file, and can be set for specific maps. All tasks are requested by other agents, with the exception of clearing a specific cell, which is used when no other tasks are requested. When a Police Force receives a task request, it calculates how long each task is going to take, and multiplies it with the weight factor, executing the cheapest one. If the cost of doing a certain task is too high, the task is not considered.

Police force agents are always active, unlike other agents that must move themselves to a certain position, in order to perform a certain action. This means that a Police Force agent, when moving, is always clearing impassable blocks in its way, and no detour of blocks is made.

Besides acting as a repeater, the Police Office computes currently assigned and unassigned Police Force tasks. The assigned tasks are multiplied by a reassign coefficient and the most suitable task for each Police Force is chosen, based on relative position to objectives. As one can easily perceive, the Police Office is, when present, the main responsible for Police Force strategy.

V. RESULTS AND CONCLUSIONS

The simulator package is an incredible piece of software, opening a broad range of research challenges in the area of heterogeneous Multi-Agent Systems. These challenges scale from the the individual decision, communication, and world model update of each agent to the coordination of the whole Rescue Team. Several teams are using learning algorithms to tackle some of these challenges. The simulator's proximity to reality is notable, streets and buildings have three dimensions, different building materials are in use and some of the sub-simulators like the fire simulator have high realistic levels.

The proposed strategical layer is now fully integrated with our rescue team and is successfully being used. The layer also maintains full compatibility with all our RoboCup soccer teams [11, 12], as the soccer model is a particular case of the specified generic layer. Results in international competitions for both the domains tested, proved the success of the layer. It as also been applied in the field of collaborative networks [13].

The model flexibility enables using it in an environment where a single program manages all homogeneous "dummy" robots, to its collective use in heterogeneous, multi-agent systems. In fact, when domains have similar nature like in soccer simulation and soccer robotic leagues, the strategies defined in one, can easily be adapted to the others. This is achieved by only modifying the roles in the existing sub-tactics.

FC Portugal's rescue simulation team successfully qualified and participated in the world cups RoboCup Osaka 2005 and RoboCup Bremen 2006 and european cup RoboLudens Eindhoven 2006. FC Portugal's rescue team won the last mentioned tournament.

ACKNOWLEDGMENTS

This research was supported by FCT under grant POSI/EIA/63240/2004. The authors wish to thank the Portuguese National Science and Technology Foundation for the fellowship granted. The authors also wish to thank LIACC, IEETA, FEUP and the University of Aveiro for their support to the project.

REFERENCES

[1] Reis, L. P., Coordination in Multi-Agent Systems: Applications in University Management and Robotic Soccer. PhD. Thesis, Univ. Porto, 2003.

[2] Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., et al., "RoboCup Rescue: search and rescue in large-scale disasters as a domain for autonomous agents research", Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, 1999. IEEE SMC '99, 1999.

[3] Reinaldo, F., Certo, J., Cordeiro, N., Reis, L. P., Camacho, R., & Lau, N., "Applying Biological Paradigms to Emerge Behaviour in RoboCup Rescue Team". In C. Bento, A. Cardoso & G. Dias (Eds.), Progress in Artificial Intelligence: 12th Portuguese Conference on Artificial Intelligence, EPIA (LNCS ed., Vol. 3808, pp. 422 - 434). Covilha, Portugal: Springer-Verlag, 2005.

[4] Paquet, S., Bernier, N., & Chaib-draa, B., "DAMAS-Rescue Description Paper". In D. Nardi, M. Riedmiller & C. Sammut (Eds.), RoboCup-2004: Robot Soccer World Cup VIII (Vol. 3276): Springer Verlag, Berlin, 2004.

[5] SOS, SOS Homepage, Amirkabir University of Technology, online, available at: http://ce.aut.ac.ir/~sos/, 2005

[6] Akin, H. L., Skinner, C., Habibi, J., Koto, T., & Casio, S. L.. Robocup 2004 Rescue Simulation League Rules V1.01. from http://robot.cmpe.boun.edu.tr/rescue2004/, 2004

[7] Committee, R., RoboCup Rescue simulator manual v0.4: The RoboCup Rescue Technical Committee, 2000.

[8] Takahashi, T., Takeuchi, I., Tetsuhiko, K., Tadokoro, S., & Noda, I., "RoboCup-Rescue Disaster Simulator Architecture". Paper presented at the Robocup 2000: Robot Soccer World Cup IV. 2000

[9] Nüssle, T., Kleiner, A., & Brenner, M., "Approaching Urban Disaster Reality: The ResQ Firesimulator" (No. 200): Univ. Freiburg, Institut für Informatik, 2004.

[10] Certo, J,, Lau N., Reis L.P., "A Generic Multi-Robot Coordination Strategic Layer", First Int. Conference on Robot Communication and Coordination RoboComm 2007, Athens, 2007

[11] Reis, L. P. and Lau, N., "FC Portugal Team Description: RoboCup 2000 Simulation League Champion", in RoboCup-2000: Robot Soccer World Cup IV, Stone, P., Balch, T., and Kraetzschmar, G., Eds., Berlin, Springer's Lecture Notes in Artificial Intelligence, 2019, Springer-Verlag, pp. 29-40, 2001.

[12] Lau, Nuno; Reis, L.P. "FC Portugal - High-level Coordination Methodologies in Soccer Robotics", International Journal of Advanced Robotic Systems: Soccer Robotics, Edited by Pedro Lima, pp. 167-192, , Itech Education and Publishing, Vienna, Austria, ISBN 978-3-902613-21-9, 2007

[14] Certo J., Lau N., Reis L.P., "A Generic Strategic Layer for Collaborative Networks", In IFIP International Federation for Information Processing, Vol. 243, Establishing the Foundations for Collaborative Networks, eds. Camarinha-Matos, L. Afsarmanesh, H., Novais, P., Analide, C., Boston Springer, pp. 273-282, September, 2007, ISBN 978-0-387-73797-3