The 5Rings "Team Description Paper"

Paulo T. Silva[1], Paulo Araújo[2], Afonso Remédios[3], Carlos Lopes[4], Bruno Basílio[5], Tiago Loureiro[6], Luís Moniz[7], Helder Coelho[8]

[1,2,3,4,5] Instituto Superior de Engenharia de Lisboa, Departamento da Engenharia da Electrónica e Telecom. e de Computadores, R. Conselheiro Emídio Navarro, 1, 1949-014 Lisboa, Portugal
[6,7,8] Faculdade de Ciências da Universidade de Lisboa, Departamento de Informática, Bloco C5, Piso 1, Campo Grande, 1749-016 Lisboa, Portugal
[1]ptrigo@isel.ipl.pt, [2]paraujo@isel.ipl.pt, [3]aremedios@deetc.isel.ipl.pt, [4]24933@alunos.isel.ipl.pt, [5]bbasilio@alunos.isel.ipl.pt, [6]scorpio_lirae@yahoo.co.uk, [7]hal@di.fc.ul.pt, [8]hcoelho@di.fc.ul.pt

Abstract. In this paper we describe the approach of the 5RingsTeam to the complexity of the RoboCupRescue simulation environment. We consider two decisive contributes to that complexity: the communication scarceness and the multiplicity of tasks to be accomplished. We propose to approach the communication scarceness aspect by two different perspectives: (a) the leadership concept (both static and emergent); and (b) the information sharing at different aggregation levels. We adopted a two level's stratification of the multiplicity of tasks to be accomplished: (a) the agent (personal) level; and (b) the team (social) level. At the present stage of development, our framework supports the search and rescue of endangered civilians (ambulance team); the repair of blocked routes to refugees (police force); and the extinguishing of fires (fire brigade). The centers (police office, ambulance center and fire station) act mainly as resource dispatchers as they call for resources (e.g. a police force), in accordance to the indications received from other agents (e.g. a blockaded ambulance team) without the required skills to handle a perceived situation (e.g. a road blockade).

1 Introduction

The RoboCupRescue is a simulation project that combines human behaviors and various disaster domains such as earthquakes, fires, logistics or traffic jams, and presents them as a coherent scene [1]. RoboCupRescue involves a very large number of heterogeneous agents in a hostile context and has been undertaken to put large-scale simulations in use in the domain of search and rescue for large-scale disasters [2].

The RoboCupRescue exhibits real world characteristics and we consider that this feat unfolds the complexity subject. We consider two decisive contributes to the complexity of RoboCupRescue environment: the communication scarceness and the multiplicity of tasks to be accomplished. The communication embraces two sensorial aspects of the agent, the audio and the visual information available on the environment. Our tasks include not only the actions (primitive tasks) directly supported by the simulation environment but also predefined or dynamically planned behavior patterns (roles).

Our final goal is to achieve coordinated team work within RoboCupRescue environment complexity. In the course of our experiments we have confirmed, the intuitive belief that each task being done independently and not in coordination, contributes to amplify the effects of the disaster.

An example of uncoordinated work, taken from our experiments, begins with a deadlock when an ambulance gets blocked in a road and that same ambulance blocks a police that pretends to clear the road's blockade; this scenario gets worst when a fire ignites near by and a fire brigade intending to extinguish the fire gets caught in the deadlock; at this point things are really complicated as the deadlock is the beginning of potential traffic jam and everyone involved is endangered by the nearby fire.

Based on Morimoto's source code [3] and technical information [4] we are presently developing a framework to validate and adjust our investigation guidelines on the complexity subject.

The next section of this paper identifies the high and low priorities of our team development. Then, we identify some properties of complex systems and characterize those properties within RoboCupRescue context. We outline our vision on the leadership and knowledge sharing issues and describe the agent and team levels. Finally, we proceed to detail the current stage of our development (the framework and the implemented behavior of each agent and each center on the RoboCupRescue environment).

2 The Preliminary Hypothesis

At the current stage of our development we accept the hypothesis that, during the RoboCupRescue simulation period, the computational resources are not overloaded by the temporal complexity of the algorithms used in the agent's deliberative process. For instance, we do not pre-compute paths in order to optimize the computational cost of the routing algorithm used by the agents, and thus minimize their eventual loss of turn in simulation cycles.

This hypothesis is supported by an early experiment using RoboCupRescue environment to analyze the communicative act (CA) distribution over a concrete simulation [5]. The experiment used RoboCupRescue version 0.31 with fire, collapse, traffic and blockade simulators. The agent software used was developed by the YabAI team [3] and won the first RoboCupRescue Simulation League international competition.

The competition rules enumerate the agents used in the experiment: 1 ambulance center, 1 police force center, 1 fire brigade center, 72 civilian, 5 ambulances, 10 police force and 10 fire brigade. In the course of the simulation, all communicative acts between each agent and the kernel are registered in a file. The resulting file incorporates a total of 60679 communicative acts.

The purpose of the experiment was to characterize the contribution of each communicative act throughout the simulation. Sequentially registered communicative acts were aggregated in 2500 size groups. The Fig. 1 presents the proportional contribution of each communicative act within each group.

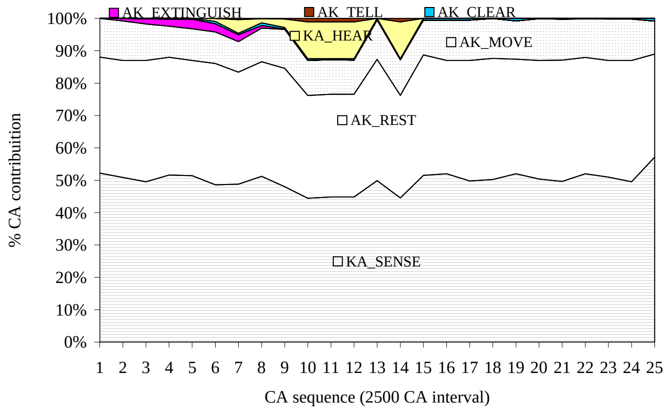

Fig. 1. Distribution of communicative acts over the simulation period.

The analysis of Fig. 1 reveals that the agents are demanding the `AK_REST` (do nothing) action for about 35% of the simulation period. The agents are idle and are not computationally overloaded; otherwise they would not be able to demand the action.

This experiment was performed in the early days of RoboCupRescue and since then we mainly kept informed about the expansion of the RoboCupRescue project. We intend to rebuild this experiment but until then we consider the temporal algorithmic complexity as a low priority development issue. The high priority issue is the acquisition of technological competence in RoboCupRescue platform in a way to exploit our investigation guidelines.

3 The Complexity of the RoboCupRescue Simulation System

The various uncertainties and costs in a real-world domain simulation system make it difficult, if not impossible, to anticipate the behavior of the system even if we completely know the function of its constituents. It is also difficult, if not impossible, to study the properties of the system by decomposing it into functionally stable parts because of the nature of their continuous interaction which even allows the system to functionally restructure itself. These difficulties match the properties of non-determinism and limited functional decomposition that characterize a complex system, as proposed by Pavard and Dugdale [6].

To get a clear-cut view of RoboCupRescue guidelines on the real-world aspect, we recall the statement of Kitano and Tadokoro [7] when discussing the RoboCupRescue simulation project, and concerning the emphasis on the following three issues: the problem dimension (over 10.000 communicating agents); the system heterogeneity (in order to survive, each agent must rely on other's capabilities); and the environment's hostile nature (every agent is in danger and communication is extremely scarce).

Soon after the first experiments with RoboCupRescue simulation system it becomes quite evident that it is impossible to anticipate the system's behavior (sometimes it is even difficult to predict a single agent's behavior). The other clear difficulty or even impossibility is the analysis of the system from the perspective of its stable parts for the simple reason that the system seems not to have any stable parts (one really feels the limited functional decomposition difficulty).

From the complexity perspective, the analysis of RoboCupRescue empirically suggests two factors that decisively contribute to the system's complexity:

- the communication scarceness, and
- the multiplicity of tasks to be accomplished.

In the course of the simulation, agents get involved in joint activities that consist of multiple tasks being performed concurrently or in sequence. According to [8], any theory of joint action should explicitly indicate when communication is necessary (each start and finish time point). The RoboCupRescue compels to relax on this explicit communication presupposition and to develop alternative approaches to achieve an acceptable joint work level.

The next sections describe the guidelines we propose to explore in order to achieve effective joint work in the RoboCupRescue competition.

3.1 The Communication Scarceness

Information is scarce because there are severe simulation rules regarding the maximum number of messages (4 messages) accepted by each agent in a simulation cycle; also inter-agent messages are not guaranteed to be delivered (UDP protocol); and it is nearly impossible for an agent to have a global and updated picture of the simulation space (both the geography and the disaster parameters). Each agent receives observations about the objects within its visible range. But there may be parts of the world that are not observable because there are no agents there.

We propose to deal with the communication scarceness from two perspectives: (a) the leadership notion as a means to introduce well-defined action guidelines and reduce inter-agent discussion and disbandment; and (b) the knowledge sharing as a way to unify the set of local views of the disaster space.

In the RoboCupRescue environment exists an institutional leadership, materialized by the command center of each agent type (police office, ambulance center and fire station). Each institutional leadership constitutes a predefined and centralized partition of the disaster space. The effectiveness of this centralized leadership depends on the reception of accurate and well-timed information sent by the agents on the field.

We consider that there is also a space for an emergent leadership that happens when apparently disparate agents adopt one agent's (the leader) goal and start working as a team. This local and short-living leadership requires low communication resources (visual communication might be enough) and its effectiveness depends on the adopted goals and on the capabilities of each team member (e.g. a team of an ambulance and a fire brigade may get blocked on their way to rescue a civilian caught in a fire, as none is apt to clear blockades).

We are working on both the institutional and emergent leadership and also on their coexistence within RoboCupRescue environment.

To deal with the knowledge sharing problem we considered the classical levels of information hierarchy: the operational; the tactical; and the strategic. The operational level is materialized by the raw sensorial data (geography, disaster parameters and messages); the tactical level results from applying aggregation operators over operational data; and the strategic level describes probabilistic tendencies.

We are already working with operational data, especially with inter-agent messages, and we plan to design aggregation operators tuned to RoboCupRescue disaster parameters.

3.2 The Multiplicity of Tasks

The multiplicity of tasks to be accomplished is layered in two levels: (a) the agent (personal) level; and (b) the team (social) level.

At the agent level the role concept [9] is introduced as a means to create and stratify sets of behavior patterns that are dynamically interchangeable and augmentable. The role concept is a design artifact that contributes to reduce the implementation complexity; it is a design unit smaller than an agent. The role is materialized as a software component, elected by an agent at execution time, and interchangeable with other roles.

The role design is centered on the minimal domain coupling requirement. The dependency between the role component and the remaining components resides on the knowledge specification, the ontology [10], and not on a specific knowledge representation technique.

The agent elects a role at its own initiative and this decision process weights the previously settled commitments (goals) at its social sphere (the team), the strength of agent's leadership dependency and the agent's social intuition on the current situation.

At the social level, the social intuition concept (the moral sense of right and wrong) is introduced as a means to reduce the space of available tasks and to promote the balance between the total social deafness and the absolute adoption of the team (embodied by its current leader) goals and plans.

Incomplete or incorrect knowledge owing to constrained sensing and uncertainty of the environment further motivate the RoboCupRescue agents to explicitly work in teams. The tasks can only be achieved by the collaborative and selfless actions of the agents [11].

The social intuition concept and the agent collaborative disposition both cross the leadership issue and we pretend to further investigate this cross join. To illustrate this idea we consider a team consisting on an ambulance and a police (leader) aiming to rescue a low injured civilian; on the way the ambulance senses a highly injured civilian and decides to immediately rescue that civilian; at that moment the team is dissolved and each agent becomes its own leader.

The agent's abstract collaborative disposition to rescue is ineffective unless properly conducted in concrete and well-defined situations. The management of the collaborative disposition is the way to achieve coordinated joint work. The power to manage (and act) is given to agents at design time and evolves during the simulation period.

As the power to manage rises also the proneness to leadership becomes higher. The RoboCupRescue command centers materialize the institutional (predefined) power to manage other's collaborative disposition. A command center loses its power (at least a certain amount) the moment it gets damaged (e.g. in fire) during the disaster simulation. The leadership disruption is unacceptable in a disaster scenario where the need to act is imperative. We pretend to look at the social intuition concept as the driving force towards coordinated work and thus the agent's induce to be a member (eventually the leader) of a team with a well-defined leadership.

4 Our Current Agents

The implementation of the previously described techniques to deal with the information scarceness is still too immature to deserve a concrete exposition.

Regarding the multiplicity of tasks issue, the activities enrolled in the agent's level are described as follows:

- elect a goal from a predefined set of goals (G);
- build a plan P (dynamically or chosen from a predefined set) to achieve the goal;
- submit each plan's primitive task (the one realized through the kernel process);
- build a memory M of the plan execution progress;
- monitor the plan execution progress (eventually using M);
- while in the monitoring stage, decide to relinquish the goal if it reveals unachievable.

We consider two subsets of G: the action goals subset Ga and the message communication subset Gc. Each decision epoch (to elect or relinquish a goal) consists of two stages, a communication stage and a goal stage. The communication stage is constrained by the RoboCupRescue competition rules. The goal stage involves the balance between the personal power to act and social authority.

At our current stage, the memory M is mainly a stack of previously visited nodes. This memory is used during the monitoring stage in order to assert on the difficulty to achieve a goal. For instance, if an agent that wants to move, keeps immobile for two consecutive simulation cycles it decides to abandon the target location and elects a previous one (where he came from). In the previous scenario the problematic location is kept in a buffer of troubled roads.

We specified the agent's communication language content (ACLC) used to share information among agents and between agents and command centers. We consider that the agent sends a message in two main circumstances. The first is when the agent is itself in danger and the second is when the agent perceives a situation beyond its own skills, or incompatible with its current goal.

The decision on what to do as a response to a message is delegated, by the centers, to the agents. We soon pretend to enforce the leadership role of centers.

The next sections materialize each one of the previous items in the context of each agent type (police force, ambulance team and fire brigade).

4.1 The Police Force and the Police Office

The Police Force agent carries out the important job of clearing the blockade roads. Also, according to the competition rules, they are twice as many as the Ambulance Teams. Thus, Police Force agents have greater mobility then other agents (as they advance in blockade roads) and they cover a larger surface due to their number. It is nearly impossible to optimize a plan to clear blockade roads as blockades appear stochastically at simulation time.

According to the Police Force characteristics its set of goals G is:

- Ga = { clear blockade roads within the surrounding area of a refugee, clear blockade roads of a largely damaged region, escape from a traffic jam };
- Gc = { *tell* (speak via telecommunication) the heard `help_me` civilian messages, *tell* and *say* (speak with natural voice) when it gets caught in a traffic jam, *tell* and *say* when it gets in danger (buried, in fire or in a road blockade) }.

The Police Force predefined plans are described in the next items along with an outlined view of each action:

- P1 ≡ get a non visited refugee; move to that refugee; get the collection of the roads within a certain radius; iterate on that collection and clear each road;
- P2 ≡ P1, except for the first action where a largely damaged region is chosen (this choice is based on perceived information and is still under consideration);
- P3 ≡ get a nearby previously visited node; move to that node;
- P4 ≡ get the location of the `help_me` message originator; *tell* the `help_me` message along with the location of its originator (or if unknown its own location). Note: the *tell* intention is kept active until the agent hears the broadcast of its own message resent through a center.

The Police Office (command center) set of goals G is:

- Ga = { keep an updated memory of road blockades and of cleared roads };
- Gc = { register newly received messages and *tell* all received messages }.

4.2 The Ambulance Team and the Ambulance Center

The Ambulance Team agent has the capability to unbury civilians, load them to an ambulance and carry them to a refugee.

The Ambulance Team agents have unique rescue capabilities but they are a scarce resource. There are few Ambulance Teams to cover the disaster space and the rescue of an injured civilian may take several simulation cycles depending on the value of the civilian's buried property.

The Ambulance Team agents have no information about the initial position of the civilians and injured people appear suddenly (e.g. by the spread of a fire) so it is nearly impossible to optimize a rescue plan.

The Ambulance Team effectiveness greatly depends on accuracy of the gathered information about the position and welfare of the injured civilians and also on the availability of unblocked roads

According to the Ambulance Team characteristics its set of goals G is:

- Ga = { rescue a civilian in a well-known position, search the surrounding buildings for an injured civilian crying for help, escape from a traffic jam };
- Gc = { *tell* and *say* when it detects a road blockade, *tell* and *say* when it gets caught in a traffic jam, *tell* and *say* when it gets in danger (buried, in fire or in a road blockade) }.

The Ambulance Team predefined plans are described in the next items along with an outlined view of each action:

- P1 ≡ get a civilian in a well-known position; move to that position; rescue the civilian; load the civilian in the ambulance; move to a refugee;
- P2 ≡ get the collection of the buildings within a certain radius; iterate on that collection and visit each building;
- P3 ≡ get a nearby previously visited node; move to that node.

The Ambulance Center (command center) set of goals G is:

- Ga = { keep an updated memory of injured civilians and of civilians already in refugees };
- Gc ≡ Police Office Gc.

4.3 The Fire Brigade and the Fire Station

The Fire Brigade agent has the capability to use fire hoses, connect them to nozzle elements, direct the fire hose, specify the water quantity discharged from each nozzle in a simulation cycle and start to extinguish a fire. In order to be effective, the Fire Brigade agent and the fire to extinguish must be within the predefined extinguishable distance. Although the effect of extinguishing a fire is not defined explicitly [4], it may not be difficult to extinguish an early fire for even a few Fire Brigade agents. On the contrary, it is difficult for even many to extinguish a late and big fire.

Time is a very important constraint as fire changes its condition rapidly and must be rapidly coped with. It is wasteful that many agents extinguish a small fire; it is ineffective that a small number of agents extinguish a large-scale fire; and it is important to circumscribe the fire by extinguishing edges of spread of fires. Also it is nearly impossible to optimize a plan to extinguish fires as there is no previous knowledge of their ignition points.

According to the Fire Brigade characteristics its set of goals G is:

- Ga = { extinguish a fire in a well-known position, search the surrounding buildings for a fires, escape from a traffic jam }
- Gc = { *tell* and *say* when it detects a blockaded road, *tell* and *say* when it gets caught in a traffic jam }

The Fire Brigade predefined plans are described in the next items along with an outlined view of each action:

- P1 ≡ get a fire well-known position; move to a fire extinguishing effective position; plug the fire hose into the nozzle elements; defined the fire hose direction and the water quantity to discharge; extinguish the fire;
- P2 ≡ get the collection of the buildings within a certain radius; iterate on that collection and look for a fire situation;
- P3 ≡ get a nearby previously visited node; move to that node.

The Fire Station (command center) set of goals G is:

- Ga = { keep an updated memory of the state of current fires and of already extinguished fires };
- Gc ≡ Police Office Gc.

5 Conclusions

Through experimentation with RoboCupRescue we already feel the strong constraints and difficulties of agent design and behavior implementation within a complex system.

We are in the process of strengthening our ideas on the leadership notion in such a dynamic environment. Also the role concept is being developed as the building block of the agent's dynamic behavior.

We pretend to achieve coordinated team work through the use of the institutional and emergent power to act and manage the other agent's predisposition to collaborative action.

Our current agents have predefined plans and simple rules to choose a plan. Each plan tries to achieve the agent's type main capability and the permanent plan monitoring tries to get the agent out of traffic jams. The command centers broadcast messages and will soon begin to proactively submit plans to their agents.

6 References

[1] Kitano, H. Tadokoro, S. Noda, I. Matsubara, H. Takahashi, T. Shinjou, A. Shimada, S. (1999). RoboCup Rescue: Search and Rescue in Large-Scale Disasters as a Domain for

Autonomous Agents Reseach, in proceedings of IEEE International Conference on Man, System and Cybernetics (SMC-99), Tokyo, JAPAN 12-15 October.
[2] Nair, R. Ito, T. Tambe, M. Marsella, S. (2000). RoboCup-Rescue: A proposal and preliminary experiences, in Proc. of ICMAS-2000 – Workshop on RoboCupRescue at IEEE Fourth International Conference on Multiagent Systems, Boston, Massachusetts, USA, 8 July.
[3] Morimoto, T. (2001). “YabAI source code”. At http://ne.cs.uec.ac.jp/~morimoto/.
[4] Morimoto, T. (2002). How to Develop a RoboCupRescue Agent for RoboCupRescue Simulation System. version 0, first edition. RoboCupRescue Technical Committee.
[5] Silva, P. T. Filipe, P. and Coelho, H. (2002). Incomplete Information Exchange in a Simulation Environment: a Communication Layer Approach. First International Joint Conference on Autonomous Agents and Multi-Agent Systems – AAMAS. Workshop on Regulated Agent-Based Social Systems: Theories and Applications – RASTA July 15-19, Bologna, Italy.
[6] Pavard, B. and Dugdale, J. (2000) An Introduction to Complexity in Social Science. In www.irit.fr/COSI/training/complexity-tutorial/complexity-tutorial.htm.
[7] Kitano, H. and Tadokoro, S. (2001). RoboCup Rescue – A Grand Challenge for Multiagent and Intelligent Systems AI Magazine. Vol 22, Iss 1, pp 39-52.
[8] Cohen, P.R. and Levesque, H.J. (1991). Teamwork. Nous, 95, 487-512.
[9] Silva, P. T. Moniz, L. Remédios, A. and Coelho, H. (2004). The Presenter Role in a Multi-agent Environment: a Knowledge Sharing Approach. Submitted to the 5th International Workshop on Agent-Based Simulation – ABS. May 3-5, Lisbon, Portugal.
[10] Gruber, T. (1993). Towards Principles for the Design of Ontologies used for Knowledge Sharing, In proc. of the Padua workshop on Formal Ontology.
[11] Nair, R. Tambe, M. and Marsella, S. (2002). Team formation for reformation in multiagent domains like RoboCupRescue. Proceedings of the International Symposium on RoboCup.