Self-Adaptive Coordination for Robot teams Accomplishing Critical Activities

Jean-Pierre Georgé, Marie-Pierre Gleizes,
Francisco J Garijo, Victor Noël, Jean-Paul Arcangeli

IRIT - Institut de Recherche en Informatique de Toulouse
118 route de Narbonne, 31062 Toulouse Cedex 4, France
{george, gleizes, garijo, vnoel, arcangeli}@irit.fr

Abstract. This paper presents a Self-Adaptive cooperation model for autonomous mobile devices, to achieve collaborative goals in crisis management scenarios. The model, which is based on the AMAS theory, allows dynamic team formation, task allocation and reconfiguration. The global behaviour emerges from interactions among individual agents. Task responsibility allocation is done by individual estimations of the degree of difficulty and priority to achieve the task. Then each peer exchanges its evaluation records with the others in order to find out the best suited peer to take the responsibility. Research work has been done in the framework of the ROSACE project. The experimental setting based on fire forest crisis management, the architecture where is embedded the Self-Adaptive cooperation model, and a working example is also described in the paper.

Introduction

Unmanned Aerial Vehicles (UAV), Unmanned Ground Vehicles (UGV) and mobile robots are extensively used in crisis management scenarios where they are in charge of achieving dangerous tasks under close human supervision. However, tight control becomes a serious shortcoming in emergency setting such as fire, where fast evolution of environmental conditions may jeopardize the safety of all actors. New generations of mobile entities helping effectively in crisis management should incorporate Multi-Agent features such as a) full autonomy to achieve individual and collective goals; b) social abilities for working as a team of mobile cooperating agents; c) self-adaptation to adjust agent's behaviour and team organization to the mission objectives by taking into account unexpected changes in the environment, internal failure and availability of mission resources.

The ROSACE[1] project faces the challenge of producing technology and tools for transforming UAVs and UGVs into Autonomous Adaptive Vehicles that are capable of cooperation to achieve collective missions in highly dynamic environments.

[1] www.irit.fr/Rosace

This paper aims to present the architecture for AAVs/AGVs and the Self-Adaptive Cooperation model that are being developed in the framework of ROSACE.

While the work on architecture focuses on structuring internal AAV complexity to facilitate abstraction, communication, and integration to process external and internal information, Self-Adaptive coordination aims at managing AAV's behaviours in order to achieve collective mission goals.

Ongoing work in ROSACE joints research efforts on MAS coordination in other domains such as (Tate 2006). In the Combined System[2] project, agents are used to implement a collaborative decision system for handling crisis situations, such as poisonous material accidental release in a city. Agents here coordinate, plan actions and reroute information of different actors from different rescue organisations. Users can also benefit from agents' information using a dedicated geo-spatial language, OpenMap, and a dedicated interaction language, Icon. Multi-agent-based Distributed Perception Networks (DPN) are also a relevant application of multi-agent systems to crisis management by intelligently aggregating information coming from a network of sensors (Maris 2006). Here these works focus on sensor and data, and the intelligence is embedded outside the devices, which implies a notable delegation to a tier computing service. Other woks, like the FP7 ALIVE[3] project and its application on crisis situations for instance (Quillinan 2009) focus more on distributed software architectures and organizational models, but dynamics and adaptation are an important part compared to more classical approaches.

The Self-Adaptive Coordination approach described in this paper is based on the AMAS theory (Gleizes, 2008) which has been applied in numerous application domains such as: mechanical design (Capera 2004), manufacturing control (Clair 2008), flood forecast (Georgé 2009), ontology creation and maintenance (Sellami 2009). Experimental results have confirmed the benefits of Self-Adaptation in open changing environments where agents have to perform quick reactions and possibly adjust their organization model in order to minimise undesirable effects and maximize system performance. However, incorporating the AMAS self-adapting model into complex physical architectures such as robots and AAVs sets up a number of scientific and engineering challenges that are being addressed in the ROSACE project: i) building a robust, flexible and efficient architecture integrating the robotics software layers and considering the specific constraints imposed to the middleware layer corresponding to the real-time embedded systems as well as the management of network resources and communication services; ii) Developing and validating decision models taking into account internal failures and hard external constraints – eg. lack of communication, immobility, lack of actuation capabilities; iii) Assessment of achievability of mission goals and distributed task allocation; iv) To guarantee the availability of communication resources for a permanent connectivity preserving the quality of communications (performance and consistency with activity requirements).

The organization of the paper is the following. After this introduction, section 2 gives a brief outline of the ROSACE experimental setting which is based on mixed teams of humans, robots and Aerial Autonomous Vehicles, cooperating for forest fire-fighting. The AMAS paradigm and its application to critical decisions is presented in

[2] http://combined.decis.nl

[3] www.ist-alive.eu

section 3 including the processing model cooperation approach and working examples. Finally the Conclusions and open issues are summarized in section 4.

The ROSACE experimental setting

The experimental setting assumes the utilization of AAV Teams and AGV Teams by a Public Organization in charge of territory supervision and fire forest crisis management. The public organization is equipped with an Emergency Management System (EMS) which provides information management and monitoring tools to achieve information fusion, and situation assessment. The EMS also provides mission management tools to help the persons in charge of the mission to asses risks, prepare the mission by recovering intervention plans, mission execution, and resource monitoring and control during mission execution.

AAVs and AGVs are situated at the intervention area to collaborate with humans for i) Location tasks, such as location of people in potential danger; ii) Supervision tasks such as fire progress monitoring; iii) Guidance to safe areas; iv) Provision of primary help to injured; v) Logistics and telecommunication support.

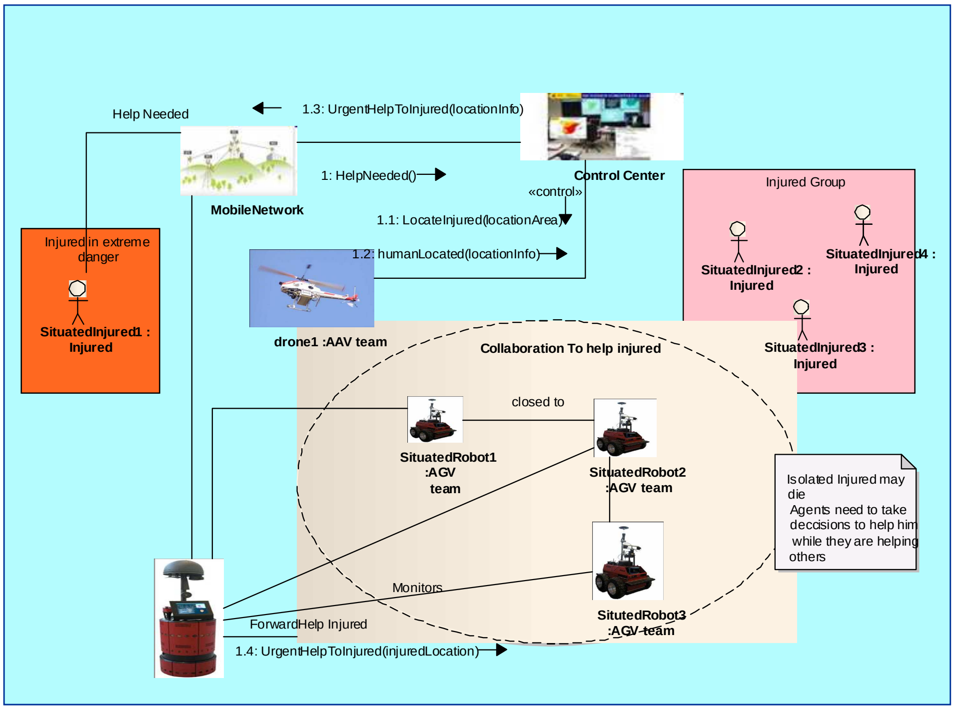

Fig. 1. Critical decision choice case TODO

A simulator has been developed to emulate environmental aspects such as the topography, flora, fauna, human presence, climatic conditions, and physical capabilities of AAVs and AGVs, such as mobility, sensing and actuation.

In order to validate robot control models a collection of operational scenarios and use cases have been defined primarily focused on agents coordination face to critical decision choices. An example of scenario is depicted in Fig. 1.

An AAV has located a group of people jeopardized by fire. Location information is sent to a team of robots that decides to go to help them. While they move to people's location a call is received in the control centre (CC) asking for urgent help. The CC orders an AAV to proceed to the injured location. Once the location has been successfully reached the CC broadcast a message asking teams close to this location to provide urgent help to this injured. The AGVs team receives this message and start deciding whether to ignore the message and continue their original task, or delegating one member of the team to help this injured. In this last case they should decide who will take the responsibility to go.

Applying the AMAS paradigm approach

The AMAS provides self-adaptation and self-organisation mechanisms for multi-agent systems in open dynamic environments. The adaptation corresponding to a change of the global system behaviour is realized by agent self-organisation. The right behaviour is reached by the right organisation of the agents. It can also be considered as the right agents acting at the right location at the right moment. Cooperation means not only that agents have to work together in order to share resources and/or competencies, but also that they should try to anticipate and avoid non cooperative situations (cooperation failures), and when non cooperation occurs, they should try act in order to reach a cooperative situation. Agents are benevolent and not altruistic in the sense of Castelfranchi (Hassas 2006). They only try to help an agent which has more difficulties than themselves if their help does not definitively prevent them to reach their individual goal.

The global behaviour at the system level emerges from interactions resulting from the agent's cooperation model. In ROSACE, the cooperation model is embedded at the decision layer. Agents are supposed to have a cooperative attitude that enables them to take decisions in a given current context, faced with new and unpredictable events. The context is defined as the agent's knowledge about itself, and about the perceived environment.

The overall architecture of the AAV where is embedded the cooperation model and a working example of implementation is detailed in the following sections.

Architectural principles. AAVs and AGVs are considered as physical agents with sensing, actuation, communication, and decision capabilities. They share the same multi-layered architecture, which is populated by manageable components offering their services to other components through standard interfaces (Fig. 2).

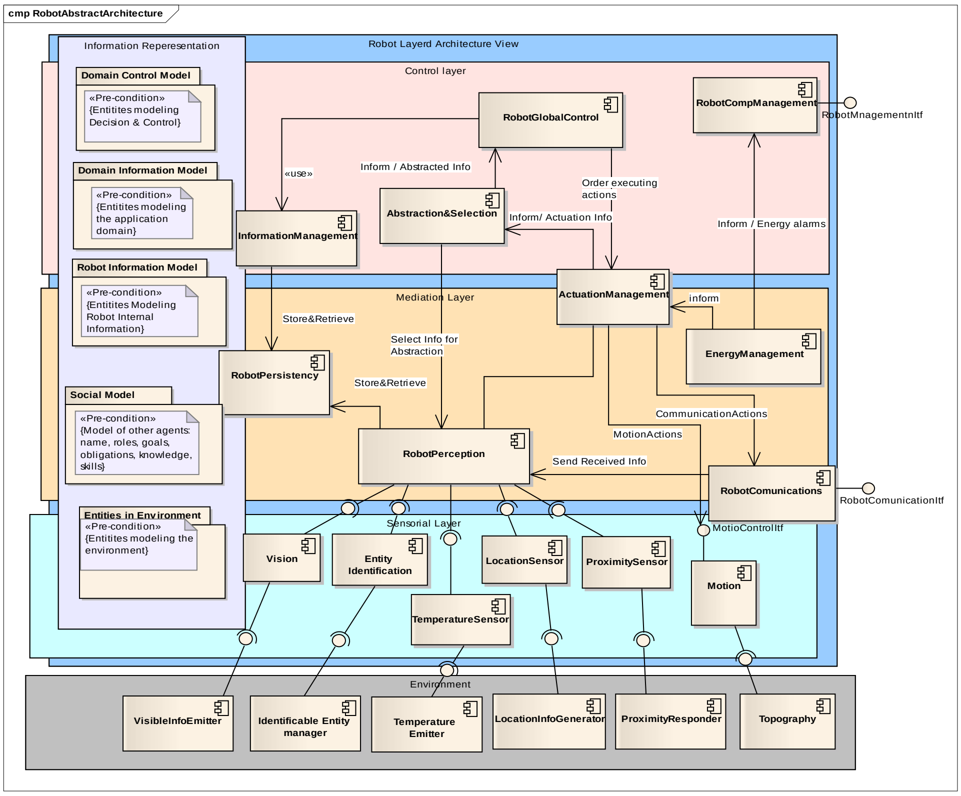

Fig. 2. General Architecture

While common components in AAVs and AGVs offer similar services, their implementation could be substantially different e.g the motion component for an aerial vehicle and for a ground vehicle.

The sensorial layer gathers the components encapsulating sensorial functions such as low level image processing, temperature acquisition, distance evaluation, obstacle detection, energy management, vision, and motion.

The mediation layer contains components that process low level information coming from the lower layer in order to simplify tasks and decisions in the control layer. The perception component aims to process, filter, select and correlate incoming information emitted by the components of the sensorial layer, as well as information received via messages sent by other agents. The persistence component provides persistence services to the upper layer. Actuation and communication components aim to provide to the components of the control layer high-level services such as moving, message sending or other actions.

The control layer is made up of the components governing the overall behaviour of the entity. The Robot Global Control Component (RGC) is in charge of orchestrating the internal components behaviours to achieve a coherent global behaviour. The RGC gathers elaborated information from the rest of the components, make choices, order execution of actions, monitor results, and send control information to relevant components when necessary.

RGC's control model is based on a declarative goal processor (Garijo 2004) that manages a goal space, and a working memory. Strategic and tactics criteria for generating goals and for selecting task to try to achieve goals are declarative. They are defined by situation-action rules, where the situation part specifies a partial state of the working memory including the objective and its internal state, and the action part contains statements for executing tasks. The processing cycle is droved by incoming information which is stored in the working memory. Then control rules are used to decide either to generate new goals, to focus on a new goal, to verify the resolution of pending goals, or to proceed to the resolution of pending goals by executing new tasks and actions.

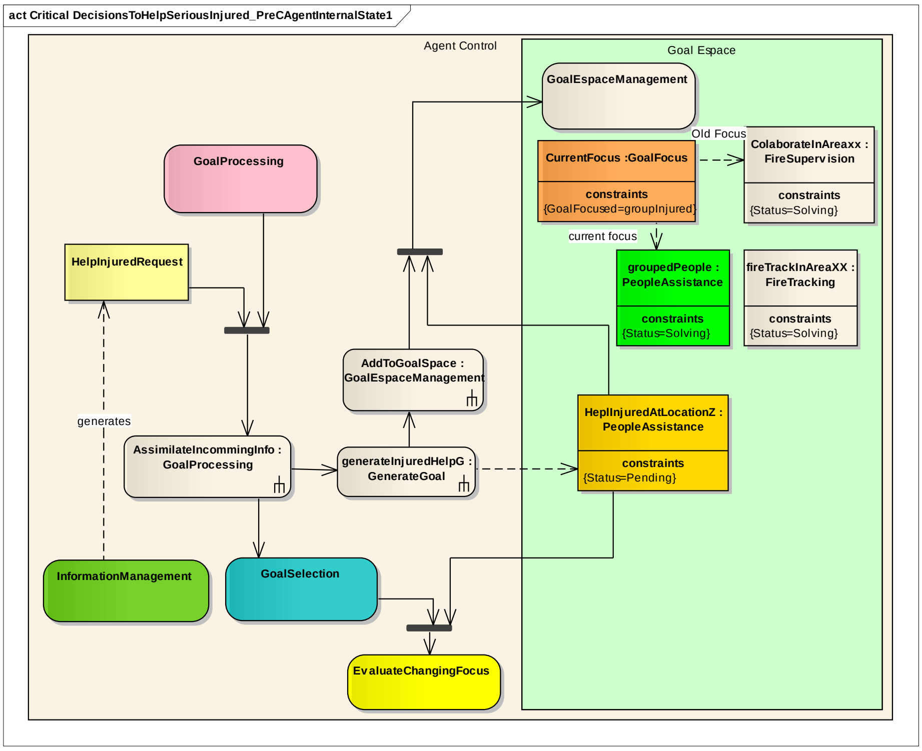

Fig. 3. Goal Processing State

The vertical layer contains information models shared by horizontal layers.

The environment in ROSACE plays a critical role since it may jeopardize the normal functioning of the whole entity. Internal components are dependent on environmental parameters such as topography or distance among networking nodes. For example the communication component is needed for coordination among cooperating peers. Decisions should then be made by taking into account both internal and external constraints.

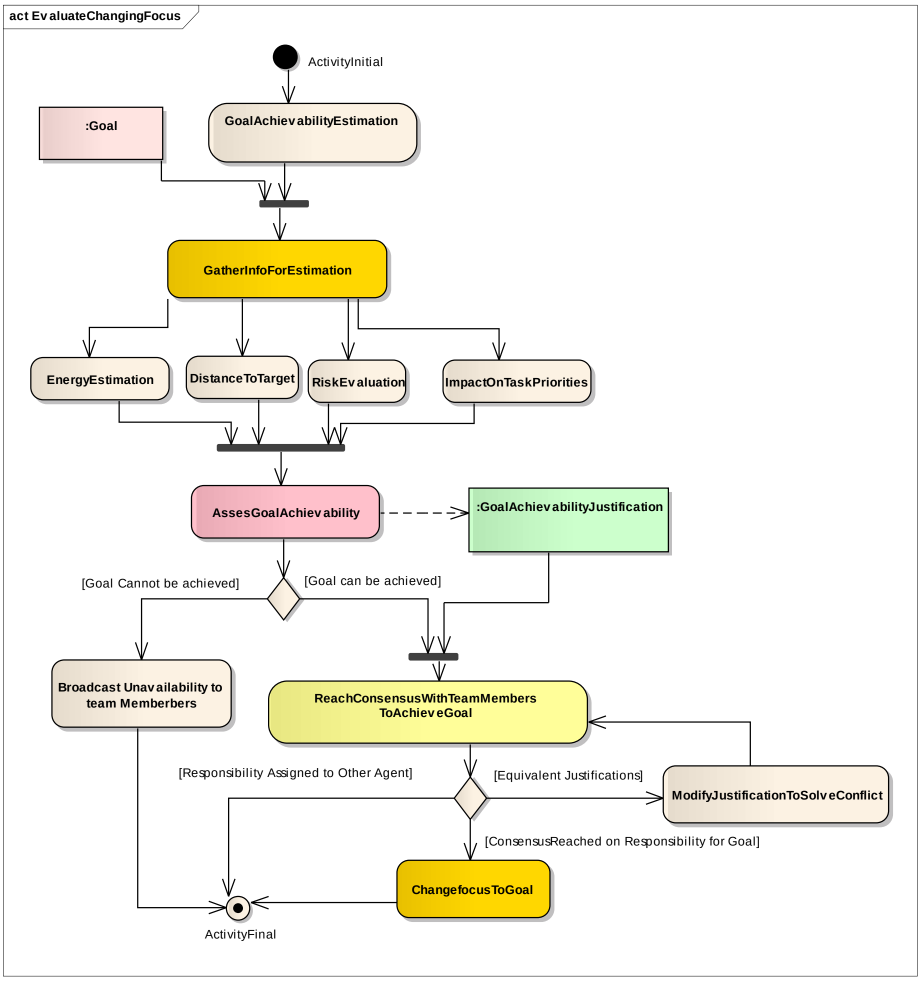

Fig. 4. Activities for Evaluating Changing focus

Decision process implementation. Decision making in AGVs Global Control is modelled as a concurrent process where the AMAS principles are applied to generate new goals in the goal space of the agent, and/or to select a goal to be achieved. The generic process for adaptive cooperation is the following:

Each agent:

- Evaluates their own capabilities to achieve the new goal
- Sends its own evaluation record to the team members
- Receives evaluation records from team members and
- Takes a decision to get the responsibility for the goal based on the best evaluation record.

Team consensus is reached when the best evaluation record exist. In this case the agent that generates this optimal record should take the responsibility to achieve the goal. When there are two or more records satisfying optimality conditions the agents which generate the records should update their evaluations to allow one of them to take the responsibility of the goal.

Working Example. While going to assist the group of jeopardized people, the goal space of each AGV in the team is focused in the same goal, which is to *provide assistance to a group of people*. The message broadcasted by the control centre is received by all the members of the team. Interpretation and evaluation of the message lead to the generation of a new goal which is *to provide assistance to an injured person at location z*. Achievement of this goal has higher priority that the goal being achieved by the team. Then a decision should be taken by the team members on who will take the responsibility of achieving this goal. Individually this means that each agent should find out evidence either to continue achieving its current goal, or to change its focus to the new goal (Fig. 3).

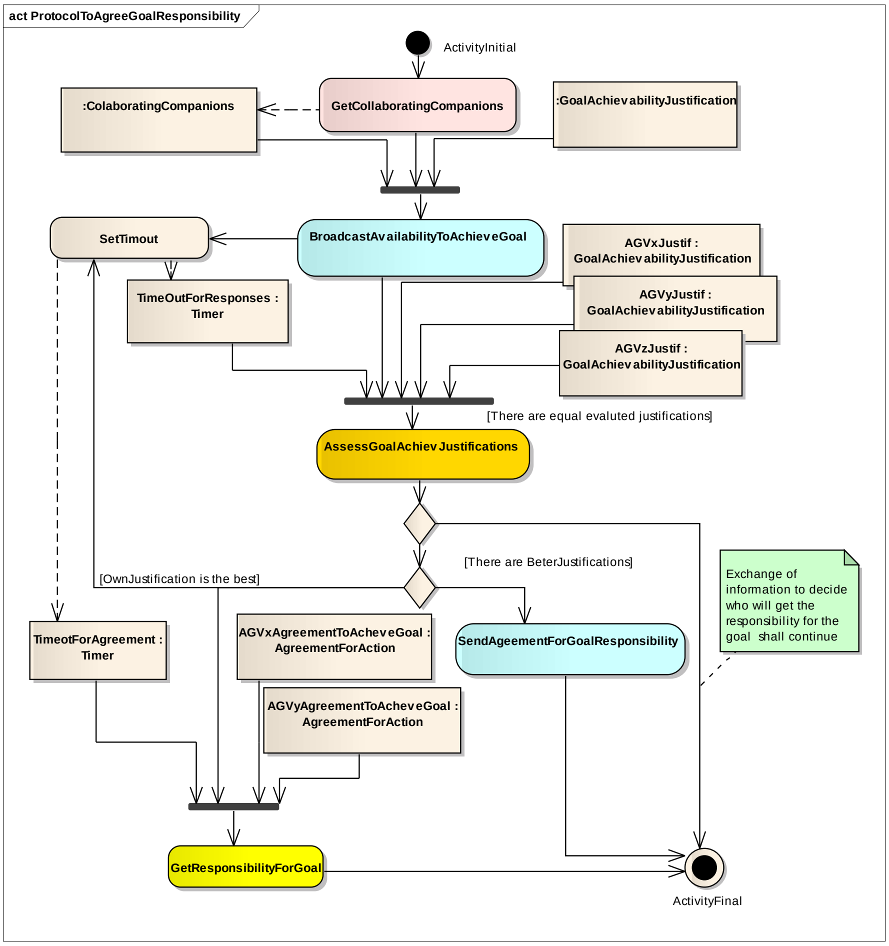

Fig. 5. Activities to reach consensus

The activity for evaluating the change of focus is depicted in Fig. 4. The first step consists in gathering relevant information (“GatherInfoForEstimation”) for assessing achievability of the goal. This information is elaborated on request by internal control tasks such as *Risk Evaluation and assessment*, and *Impact on task priorities*, and by AGVs internal components managing *energy* and *distance* through the distance sensors. This information represents the knowledge about the current context of the agent. They are stored in the Information Representation layer (Fig. 2). Then, the assessment task *AssessGoalAchievability* is to analyze if the new goal could be or not achieved alone by itself. This analysis is based on the cooperation attitude of the agent and on the local point of view it has about the situation. The agent must check its own constraints such as: have I enough energy? Has my current goal more priority? Have I all competencies?… This analysis provides a degree of difficulty to adopt this new goal or to participate to this new goal. So, the assessment task generates a goal achievability report which is used to determine whether or not the goal could be achieved by the agent and if it could the degree of difficulty for the agent to do it. If it concludes that the goal could be achieved it generates a goal achievement report which summarized the cost estimation for achieving the goal.

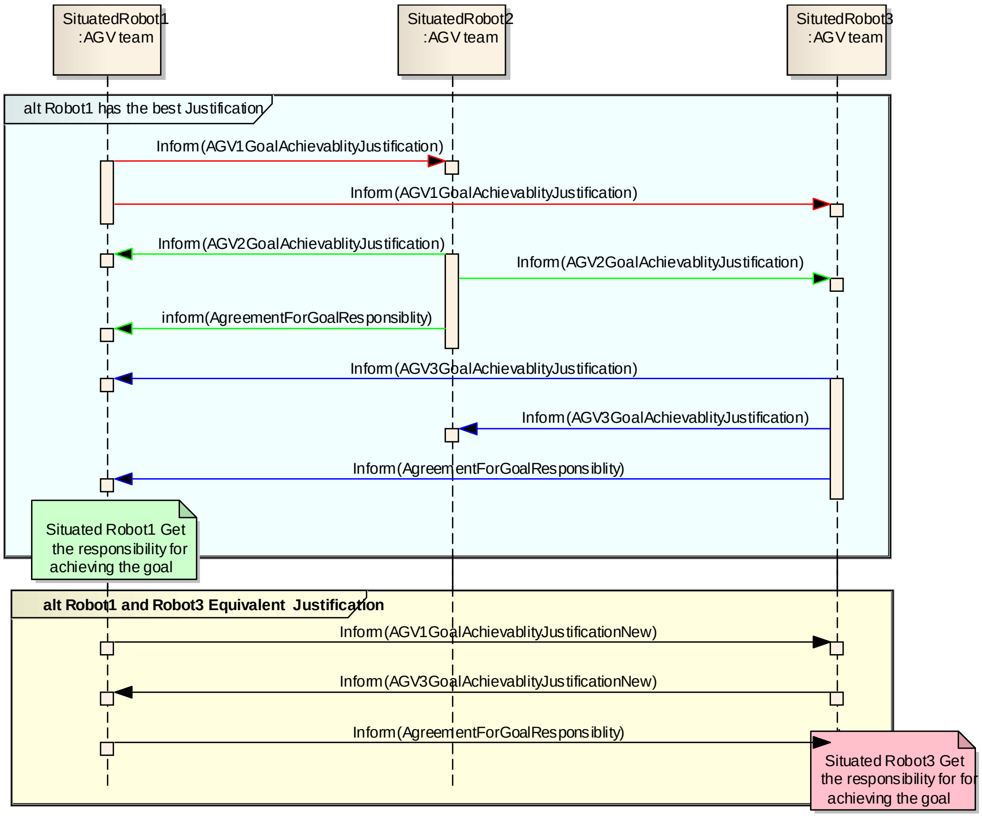

Fig. 6. Messages exchanged to get responsibility for the goal

Then, the agent has to get its neighbours to analyse its perceptions. Its neighbours can be: all agents in the perception area of its camera, all agents with which it can communicate to using its networking resources. The neighbours at a given time can be considered as the temporary team. This temporary team can change dynamically over time (for example between two perception steps).

After, all agents have to exchange their achievability report concerning a given goal, and given their cooperative attitude they will do the same reasoning with the same set of information. Details of the activity to reach consensus are depicted in Fig. 5.

So, knowing all information about the degree of difficulty for each agent, an agent executes the following algorithm: For a temporary team composed of N agents, for each tuple (Ai, Di, G) where Ai is an agent, Di is its degree of difficulty to reach the goal G and i varying from 1 to N.

The current agent Ac chooses the tuple (Rmin, Dmin, G) where Dmin = Min(Di)

for *i* $\in$ *[1..N]*

if *min = c*

then *agent Ac accepts to change its goal to provide help to others agents*

else *it does not change its current goal and continues*

The observable team behaviour as a result of the internal process described above is in detailed in Fig. 6.

Conclusions and future challenges

The self-adaptive cooperation model has been implemented in a simulated environment based on Blender[4] with a limited number of robots. In comparison to more sophisticated cooperative models based on agent's teams where agents have fixed roles, and have complex decision making mechanisms, the proposed solution is simple, easy to implement and efficient.

Self-adapting agents are capable of assimilating the changes in the environment for improving the achievement of its tasks, and also for making decisions to assume mission tasks taking into account the point of view of its cooperating peers. As agents have common mechanisms to avoid non cooperative situations, possible conflicts which could block task allocation are minimised or deleted. Tasks are assumed by the best situated agent, then ate the global level cooperation succeeds.

While initial results seems promising, research work should continue to assess model performance taking into account scalability issues as well as internal components failures and external constraints such as lack of communications, week energy, immobility, uncertainty of perceived data, and others. At the theoretical level formal demonstration of the effectiveness of evaluation functions for task allocation and decision making is also necessary.

[4] www.blender.org

Acknowledgments

This project is supported by the RTRA STAE[5] foundation.

References

(Capera 2004) Capera D., Gleizes M-P., Glize P.: Mechanism Type Synthesis based on Self-Assembling Agents. Dans / In : Journal of Applied Artificial Intelligence, Taylor & Francis Group, Vol. 18 N. Numbers 9-10, p. 921-936, octobre / october 2004.

(Clair 2008) Clair G., Kaddoum E., Gleizes M-P., Picard G.: Self-Regulation in Self-Organising Multi-Agent Systems for Adaptive and Intelligent Manufacturing Control. In : IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008), Venice Italy, 20/10/2008-24/10/2008, IEEE Computer Society, 2008.

(Garijo 2004) Garijo, F., Bravo, S., Gonzalez, J., Bobadilla, E. (2004) BOGAR LN: An agent based component framework for developing multi-modal services using natural language. LNAI 3040, pp. 207-220.

(Georgé 2009) Georgé J-P., Peyruqueou S., Régis C., Glize P.: Experiencing Self-Adaptive MAS for Real-Time Decision Support Systems. In : International Conference on Practical Applications of Agents and Multiagent Systems, Salamanca (Spain), 25/03/2009-27/03/2009, Yves Demazeau, Juan Pavon, Juan M. Corchado, Javier Bajo (Eds.).

(Gleizes, 2008) Gleizes M-P., Camps V., Georgé J-P., Capera D.. Engineering Systems which Generate Emergent Functionalities. Dans / *In* : Engineering Environment-Mediated Multiagent Systems - Satellite Conference held at The European Conference on Complex Systems (EEMMAS 2007), Dresden, Germany, 01/10/2007-05/10/2007, Danny Weyns, Sven Brueckner, Yves Demazeau (Eds.), Springer-Verlag, Lecture Notes in Artificial Intelligence (LNAI) 5049, *july* 2008.

(Hassas 2006) Hassas, S., Castelfranchi, C., Di Marzo Serugendo, G., A., K.: Self-Organising Mechanisms from Social and Business/Economics Approaches. Informatica 30(1) 2006.

(Maris 2006) Maris M., Pavlin G.: Distributed Perception Networks for Crisis Management. In Proceedings of the Third International Conference on Information Systems for Crisis Response and Management (ISCRAM 2006), New Jersey, USA, May 2006.

(Quillinan 2009) Quillinan T. B., Brazier F., Aldewereld H., Dignum F., Dignum V., Penserini L., Wijngaards N.: Developing Agent-based Organizational Models for Crisis Management. In Proceedings of the Industry Track of the 8th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2009), Budapest, Hungary, 2009.

(Sellami 2009) Sellami Z., Gleizes M-P., Aussenac-Gilles N., Rougemaille S.: Dynamic ontology co-construction based on adaptive multi-agent technology. In : International Conference on Knowledge Engineering and Ontology Development (KEOD 2009), Madeira (Portugal), 06/10/2009-08/10/2009, Springer, 2009.

(Tate 2006) Tate A.: The “Helpful Environment”: Geographically Dispersed Intelligent Agents That Collaborate, Vol. 21, No. 3, IEEE Intelligent System, May/June 2006.

[5] www.fondation-stae.net