Role-based Dynamic Coalitions of Multi-Tasked Rescue Robots

Van Tuan Le
Université de Lille-Nord de France
Ecole des Mines de Douai
tuan@hanoiscrum.net

Noury Bouraqadi
Université de Lille-Nord de France
Ecole des Mines de Douai
noury.bouraqadi@mines-douai.fr

Serge Stinckwich
UMMISCO, UMI 209, Hanoi, Vietnam
GREYC, UMR 6072, Caen, France
serge.stinckwich@ird.fr

Arnaud Doniec
Université de Lille-Nord de France
Ecole des Mines de Douai
arnaud.doniec@mines-douai.fr

ABSTRACT

Organizations allow structuring and coordinating the activities of robots that take part in a multi-robot system (MRS). Within a given organization, each robot is assigned to a role that governs its behavior and its interactions with the other members of the MRS. In this paper; we investigate in a class of problems where role allocation must be done dynamically. This applies, for example in the context of rescue robotic applications where neither the number of robots nor characteristics are known *a priori*. Furthermore, tasks to be performed are not necessarily all known or at least a portion of the information remains to be discovered (e.g. locations of victims). Finally, some robots may temporarily leave the MRS (for battery recharging) or permanently due to failure or breakage. We propose a solution that can dynamically allocate roles to robots and revise the allocation. This revision takes place in case of failure of agents or in case of discovery of a new task. This allocation allows agents to participate in several tasks.

Keywords

Search and rescue robotics, team coordination, dynamic coalitions, multi-tasked robots

INTRODUCTION

In many applications such as robotic rescue, MRS designers have little information on robots to be used. The number of robots and their individual skills may vary from one mission to another. From there, the organization of the MRS should be defined in an abstract manner. The concept of role can meet this need for abstraction. The need remains to allocate roles to robots. This process may be tedious, especially as the number of robots may be very large and heterogeneous. Furthermore, in the dynamic environment usually found in search and rescue missions, tasks can be perceived by robots and the online allocation of roles (as tasks are discovered) may be more relevant. Indeed, the choice of robots to play a role may take into account the context of the task (e.g. its position), robots' resources and their possible involvement in other tasks.

In this paper, we propose a solution that addresses the dynamic formation of overlapping coalitions with heterogeneous robots. Each discovered task is treated by a coalition of robots formed dynamically. The coalition is a set of robots recruited on the basis of bids, to play certain roles. Coalitions may potentially overlap the sense that they share robots. Indeed, robots can participate in the completion of several tasks at the same time and therefore assume roles in different coalitions.

The paper is organized into 5 parts. In section COOPERATION CONCERNS, the notions of role and task are compared in the light of a simple search and rescue example involving 3 robots, this section is then followed by a brief review of RELATED WORKS. Our proposal for dynamic coalition formation is described in PROPOSITION. In Section 4, a complete search and rescue example highlighting the benefits of the approach is introduced. Finally Section CONCLUSION closes the paper.

COOPERATION CONCERNS

Reference Scenario

The cooperative task is to find victims in a disaster area. Victims positions are unknown by the robots and they need to explore the area in order to determine their positions. When a victim is discovered, they should be transported to a shelter by a robot if the path has been cleared before. To carry out this task, we need to have two autonomous mobile robots, namely a **Transporter** and a **Path Clearer**. In addition, an unmanned aerial vehicle (UAV) could provide images from the disaster area in order to locate victims in the area. The **Transporter** with neither visual sensor nor tracking capabilities can as its name suggest transport the victim. It needs to be informed in real time about its location and its position relative to the victim. In order to remove obstacles, the **Path Clearer** must clear the path while the **Transporter** moves the victim to a shelter. As such, the group of three robots works well until the **Transporter** moves the victim outside the camera's coverage area. The UAV is no longer able to provide instructions for locating the **Transporter**. Thus to continue the task, it should be replaced. The **Path Clearer** if he has some cameras is a good candidate to guide the **Transporter**.

Mission, Task and Sub-tasks

We call mission the overall objective that must be accomplished by a group of robots. A mission can be decomposed into a set of tasks. A task is a part of a mission that can be performed independently from the rest of the mission. Each task can in turn be decomposed into sub-tasks. This subdivision continues until we reach an elementary sub-task. A (sub-)task is called elementary if it is implemented directly by a single robot. It is clear that the decomposition of a task into subtasks depends firstly on robots' capabilities.

Once the decomposition carried out, the elementary tasks are distributed over the robots. The degree of coordination of robots depends on the degree of interdependence of sub-tasks. This interdependence must be explicitly expressed in the definition of sub-tasks. For example, our scenario describes a mission that involves only one single task, subdivided into three sub-tasks: 1) localize the victim 2) transport the victim to a safer place, and 3) keep the path clear. These three sub-tasks are interrelated: a robot dealing with sub-task 2 depends on robots in charge of sub-tasks 1 and 3.

Role vs. Task

The organization of a MRS is often defined using in particular the concept of role. The behavior of each robot is driven by the role it plays in the system. However, the mission of a MRS can be decomposed into task as stated earlier. Thus, the question of the relationship between role and task arises naturally. There are many definitions related to these two concepts (G. Thomas and A. B. Williams, 2005; Gerkey and Mataric, 2004). In this paper, we consider that a task is a unit of work, which often must be completed within a certain period of time. The completion of a task requires the execution of a set of actions whilst a role denotes the function assured by a robot within a MRS, that is to say, the robot behavior and relationships with other robots of the system. Thus, the role played by a robot defines tasks that the robot can/should realize or not.

Our proposal is based on the concept of role. Indeed, it enables development of generic organizations that is to say independently of robots. Based on the mission of MRS, the designers define roles that must be fulfilled by robots. These roles can be either played by a single robot or they can be spread over different robots. But, from a logical perspective, interactions between these roles are always the same. In our previous example, we can sketch 3 roles: **Transporter** – corresponds to the behavior: transporting a victim; **Path Clearer** – corresponds to the behavior: moving obstacles; and a **Pilot** – corresponds to the behavior: locating objects and disseminating guidance information. These three roles can be distributed over 3 different robots. They may even be played by a single robot.

RELATED WORKS

(Stone and Veloso, 1999) proposes a solution that applies to problems where a team requires periodic synchronization. Team is arranged in formation which includes a set of roles, grouped into units. Each role is taken by a single robot and a robot can take only one role. The robot will play a role in a formation fixedly determined. A robot does change its role only when the team moves from one formation to another. These changes are made by all robots during pre-determined appointments dedicated to the synchronization between robots. The team then decides the formation change based on performance indicators shared by all members of the team. Each robot in the team is aware of the chosen formation and therefore the role it should play. Indeed, the roles of each formation are pre-assigned to robots. This is the two limitations to this work: the solution does not take into account the context of execution for each robot and the number of robots and roles are identical. The solution cannot take advantage of the arrival of new robots and does not take into account possible failures of participating robots.

RACHNA (Vig and Adams, 2008) is a solution that can form coalitions of any size, but a robot can participate in only one coalition at a time. In RACHNA, there are Service Agents managing a set of agents and each provides a particular service (Foraging Service Agent, Pusher Service Agent, etc.). Each task is managed by a Task Agent. To form a coalition, a Task Agent has to pay Service Agents that manage agent suppliers of services required. Each Service Agent who accepts the offered wage, selects from the agent that it manages and assigns roles in the coalition to these agents. Thus, contrary to other work, the auction process in RACHNA is reversed. It is the Task Agent which sets a price to buy a particular service, whilst the Service Agents decide if they agree to sell only the services provided by the robot they manage to the Task Agent.

PROPOSITION: DYNAMIC COALITION FORMATION

Hypothesis

Our proposal requires having an organization predefined by the MRS' designers before the beginning of the mission. This definition describes coalitions that robots may form to perform each type of task that can appear during the mission.

At the beginning of MRS deployment, each robot has the definition of the organization, but no effective coalition is formed. The formation of coalitions and therefore the allocation of roles are performed dynamically and online by robots themselves based on the task they find. A robot takes part in a coalition if he plays at least one role in this coalition. Moreover, a robot can play several roles in a coalition or even in several coalitions.

The assignment of roles requires that robots can communicate. In the context of applications using mobile robots, we make the assumption that a network-connectivity-maintaining algorithm (Le, Bouraqadi, Moraru, Stinckwich, and Doniec, 2009) is implemented.

Constraints on Role Assignment

To be able to play a specific role, a robot must *a priori* possess knowledge and skills required for this role. Therefore, before participating in a coalition, a robot is able to identify roles that it can potentially take on. We call such roles compatible roles with the given robot.

Moreover, robots can possibly play several roles simultaneously. But, these roles should be compatible with each other. That is, they should not introduce contradictory behavior. We call this restriction concurrency constraint. It prohibits a robot from playing simultaneously some combinations among its compatible roles.

Coalitions Specifications

A coalition C_i is specified by (*roles, constraints, taskType*) where *roles* is the set of roles $\{r_1; r_2, ..., r_{li}\}$, and constraints of the coalition is a set of constraints $\{c_1, c_2, ..., c_{li}\}$ on roles. Each constraint c_k for a role r_k is of form $(min_{rk}; max_{rk})$ and states the minimum (min_{rk}) and the maximum (max_{rk}) number of robots that should play the role. Last, *taskType* refers to a family of tasks which can be addressed with coalition C_i.

Specifications of coalitions are held by robots which play the role of *Coalition Manager*. A robot that plays this organizational role (see section below) serves as the auctioneer in the bidding process for the formation of coalitions.

Roles specifications

All specifications of roles in an organization are defined independently from coalitions. Indeed, the same specification of a role can be used in different coalitions.

A role *r* is specified by the tuple (*id, concurrents, utility, behSpec*). The first element *id* is the unique identifier of the role, not only in the coalition, but also in the whole system. It is used in communications for the formation of coalitions to identify roles unambiguously. The *concurrents* is the set of identifiers of roles that are concurrent to *r*, that is, roles that cannot be played by a robot which plays role *r*. The *utility* is the definition of the utility function of the role. It allows a robot to calculate the benefit it would get after accomplishing duties of role *r*. Finally, *behSpec* is the specification of the behavior associated with the role. This includes both actions that can be achieved by the robot and interactions it may have with the other robots of the MRS. The specifications of the roles are stored on the robots at the beginning of the mission. Indeed, each robot has the specifications of the roles that are compatible.

Finally, note that a role is specific to a given coalition. Two coalitions can specify the same behavior with roles that have different IDs. Indeed, a robot that plays a particular role in a coalition (e.g. pushing a box) can not necessarily provide the same time the same service in another coalition.

Organizational roles

To ensure a completely decentralized control of the MRS, we use roles that we refer to as organizational. These roles are used to manage the organization and deal more specifically with the management of coalitions' lifecycle.

- *Task Detector* This role is simply to detect some type of tasks and announce them. The announcement provides the type of the detected task along with contextual information, such as: geographical coordinates of the place of intervention or the degree of urgency or severity of the task. A robot playing that role may eventually be asked to provide additional information (such as a photo of another view, or a new measure of temperature) about the task. Note that depending on the missions, there may be different roles of Task Detector if tasks to detect are different in nature. For example, fire detection requires different sensors and equipment as to those required for detection of gas leaking or flooding.
- *Coalition Manager*. A robot that plays this role triggers the formation of a coalition and then coordinates it when it receives the announcement of a new discovered task. It sends a message to other robots to participate in auctions on the roles of the new coalition. It then manages the auctions of potential participants, allocates roles and frees robots once the task has been accomplished. It may also decide to cancel the formation of the coalition if all roles are not allocated. Finally, it manages departures or failures of participants in the coalition and eventually replaced.

Regarding the allocation of organizational roles, we consider that an optimal strategy may vary depending on applications. In some cases it may be worth allocating the roles of task detector to as many robots having adequate sensors. But the proliferation of sensors generates a potentially large number of advertisements for the same task. A *Coalition Manager* must be able to decide if two advertisements are redundant or not.

The number of robots who take on the role of *Coalition Manager* should depend on the maximum number of coalitions that can be formed simultaneously. That is the maximum number of tasks that can be treated simultaneously. It also depends on the nature of robots and the nature of communication networks. Thus, robots communicating via a wireless interface can be divided into different disconnected networks. Nevertheless, robots of each network must have at least one *Coalition Manager*.

Finally, note that the multiplication of Coalition Manager creates itself another allocation problem. Upon appearance of a task, they must decide together which one is responsible for managing the coalition to address the detected task.

Auction-Based Role Assignment

The solution we propose to allocate the roles of a coalition is based on auctions. Once a task is detected, the *Coalition Manager* announces the opening of bids by providing the identification of the roles to be bided on. Each other robot *R* may issue bids on all every role *r* of the coalition that is compatible with *R* and that does not violate the concurrency constraints with roles played by *R*. The auction is computed using the utility function that is specific to the role. Thus, all robots bidding on a same role use the same utility function. This bidding value takes into account resources of the robot and its context. For example, two identical robots bidding on the same role may submit different bids because of their potentially different states: level of battery, distance to the task, etc. The identical auctions will be cleared by the *Coalition Manager* based on their order of arrival.

After a certain period of time, the Coalition Manager executes the allocation algorithm 1. If it finds a solution, it then sends to each robot R_i identifiers of roles that are assigned to R_i. The activity of the coalition can start when enough robots confirm the acceptance of roles assigned to them, that is to say when the constraint on the minimum number of robots per role in the coalition is verified.

Algorithm 1 assigns roles to robots in two steps. The first step (lines 2 and 7) isolates the possible allocations for each role. The second step (lines 9 to 14) selects for each role as many robots as possible.

```
input  : C, bids_C.
/* C: specification of a coalition.
/* bids_C: set of bids for all roles in C.
output: Set of valid assignments VA in case of sucess, ∅ otherwise
1  begin
2      foreach r ∈ C.roles do
3          bids[r] ← ∅;
           /* bids[r] : set of possible assignments of role r
4      end
5      foreach bid ∈ bids_C do
6          bids[bid.role] ← bids[bid.role] ∪ {bid};
           /* bid.role denotes the role which bid is bid
7      end
8      VA ← ∅;
9      foreach r ∈ C.roles do
10         if | bids[r] |< min_r then
11             return ∅;
12         sorting bids[r] in order of decreasing bid;
13         VA ← VA∪ {the min(| bids[r] |, max_r) first elements in bids[r]};
14     end
15     return VA;
16 end
```

Algorithm 1: *Optimistic* Assignment with Pre-Filtering

CONCLUSION

We have presented in this paper a solution for the dynamic allocation of roles to a team of rescue robots. The organization is described by specifying a set of coalitions. Each coalition is intended to address a particular type of task. Robots have the responsibility to detect tasks and to select and form appropriate coalitions. The recruitment process for a coalition is based on a bidding process with pre-filtering. Every robot that wants to participate in a coalition places a bid only on roles that it "wants" to play. These roles are both compatible to its skills and resources and are not concurrent with the roles it already plays. In addition to giving more autonomy to the robots, pre-filtering distributes the computational load on the different robots and reduces the number of messages exchanged with the *Coalition Manager*. The latter shall check the cardinal constraints of each role. These constraints are defined in the description of the coalition and refer to the minimum and maximum numbers of robots that should play each role.

This work revealed the need for organizational roles essential for the management of the organization. A perspective would be to study the deployment strategies of these roles. A further question is the integration of a network management strategy for wireless communications as part of our robotics application [5].

REFERENCES

1. G. Thomas and A. B. Williams. Roles in the context of multiagent task relationships. in Proceedings of AAAI Fall Symposium "Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems,", TR FS-05-08, ISBN 978-1-57735-254-9., 2005.
2. B. P. Gerkey and M. J. Mataric. RoboCup 2003: Robot Soccer World Cup VII, vol. 3020/2004, ch. On role allocation in robocup, pp. 43–53. Springer Berlin / Heidelberg, 2004.
3. V. T. Le, N. Bouraqadi, V. Moraru, S. Stinckwich, and A. Doniec. Making networked robot connectivity-aware. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 3502–3507, 2009.
4. P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. Artificial Intelligence, vol. 110, no. 2, pp. 241–273, 1999.
5. L. Vig and J. A. Adams, Coalition formation: From software agents to robots. Journal of Intelligent Robot System, vol. 50, no. 1, pp. 85–118, 2007.