A Context Aware Multi-robot Coordination System based on Agent technology

Alberto Grosso, Davide Anghinolfi, Antonio Boccalatte, and Giorgio Cannata
DIST
University of Genova
Genova, Italy
alberto.grosso@unige.it

Abstract—**This paper presents an approach for multi-robot coordination based both on coordinated navigation and task allocation method. An ad hoc agent based architecture is defined in order to implement the robot control system in both simulation and real applications. The coordination of the multi-robot system is based on agent interaction and negotiation, and a communication infrastructure based on open web standards is provided. The system employs the RFID technology for building a context aware information system which is the base of the coordination strategies.**

Multi-Robot Systems; Multi-Agent System; Distributed Interactive Systems; Multi-Robot Task Allocation.

I. INTRODUCTION

The general objective of this work is to develop an open knowledge environment for self-configurable, low-cost and robust robot swarms usable in everyday applications. Advances in the state-of-the art of networked robotics are proposed through introduction of a local and global knowledge base for ad hoc communication within a low-cost swarm of autonomous robots operating in the surrounding smart IT infrastructure.

The work will address the development of flexible, cost-effective, dependable, and user driven robot swarm, which possesses a higher intelligence collectively than each member of the swarm independently. In particular, this paper involves self-organizing task-sharing mechanisms between individual members of the swarm to capitalize on the availability of a large number of simple and cost-efficient robots. Reuse of the local and global level knowledge is pursued by creating on-site (near to objects of interest) distributed data environment. System is kept scalable and manageable by decentralization of control and manipulation to local level.

In this work RFID tags [4, 5] are used for deploying information through the environment that is used for applying distributed coordination strategies. It is assumed that RFIDs are distributed in the environment prior to robot operation, and used to build a sort of “navigation graph”: each RFIDs contains navigation directions to neighboring RFIDs, thus allowing robots to safely execute paths from and to different locations. We decide to exploit the localized information tagged on RFIDs at different levels:

- to drive the swarm through the environment: the tags make it a smart area;
- to allow local communication;
- to provide the infrastructure for constituting context aware knowledge.

The proposed system is based on both a coordination control algorithm and a distributed task allocation method: as mentioned above, both methods rely on RFID technology and exploit context aware information for navigation and task execution aims. An ad hoc agent-based architecture is defined for implementing the robot control system. AgentService [14] is adopted for abstracting hardware level and making the developed software work both in simulation and in real application, a previous implementation of the architecture was based on Player/Stage tool [19] and detailed in [20].

In next paragraphs, the state of the art in applying RFID technology and task allocation methods to multi-robot systems is briefly analyzed. Section 2 illustrates a novel approach for robot team coordination that could be applied in different application contexts; in addition we present the agent-based architecture implemented for applying the coordination methods in robot applications. Conclusions are given in Section 3.

A. *Exploiting RFID technology in the field of multi-robot system*

RFID tags have attracted great attention as a key device in various domains, from object tracking to robotics. As a result, the deployment of RFID technology on a larger scale is about to become both technically and economically feasible [6].

Focusing on mobile robotics applications, attached to walls, machines, or other specific places in the environment, RFID tags works as dedicated landmarks making the robot able to detect items, obtain information about its position, and even get instructions to reach a given goal [5]. Considering the analysis of the state-of-the-art, different aspects are investigated in literature obtaining different level of success. The main topics considered by researchers in applying RFID technology to robotics are: localization, coordination, exploration, and object recognition [1, 7, 18]. The basic idea of such approaches is to employ the wireless sensor network, provided by RFID technology, as a way to coordinate teams of robotic agents and human by employing its ad-hoc communication capability in addition to its sensing. Moreover, novel approaches are based on the concept of stigmergy: a swarm of agents can coordinate its components' behavior by modifying the environment [8, 9].

B. Multi-robot Task Allocation Algorithms

Sharing the task related workload among robots through a task allocation mechanism is an effective way for making a team of robots act in a coordinated way. A task can be decomposed into sub-tasks and an assignment process can be adopted in order to allocate tasks among the team. Different approaches are proposed in literature for solving the task allocation problem starting from classical operational research algorithms to social behaviors inspired by human society or swarm organization.

One of the most diffused and adopted mechanism for task allocation are auctioning systems [10]. They naturally form a distributed framework, and offer a good way of sharing information between agents/robots. The basic idea behind a market-based task-allocation mechanism is to assign tasks via an auction, where agents/robots bid a value in a shared currency based on their perceived fitness for a task [10]. Tasks are awarded to the lowest bidder if the goal is minimizing cost, or to the highest bidder if the goal is to maximize reward. One first work is M+ [11] but a lot of studies are presented, papers in [12, 13] illustrate some interesting results.

II. The Context Aware Multi-robot Coordination System

A. The Context Aware Multi-robot Coordination System

Cooperative robotics is one of the hottest topics in current research, as shown by the growing number of articles published every year. There are two main ways of achieving coordinated multi-robot behaviors: making the single robot control processes work coordinated; orchestrating single robot behaviors by allocating properly the tasks to be accomplished. The former approach operates by defining robot control architecture based on algorithms which take into account the behaviors of other robots belonging to the team. In the latter, robots compete or cooperate for acquiring tasks, or a central system allocates the tasks optimizing some parameters; after that, each robot performs the assigned tasks as a single system.

We think that the two approaches are complementary and can be applied together in order to obtain a whole coordinated behavior of the robot swarm. We selected for our multi-robot application a classical problem that can be applied in different context. A traditional problem in multi-robot system is to repeatedly visiting a set of specific locations, a variant of the dynamic coverage problem (see [1] and the reference within) which is particularly relevant for surveillance and patrolling, and – to some extent – for repeatedly cleaning crowded areas. In addition to this general problem, we choose to add the possibility to assign to the robots' team context specific tasks such as guard a specific area, clean a zone, etc.

We propose a coordination system in which the coverage problem is solved with a real time search algorithm presented in [15]. Each robot of the swarm performs the coordinated coverage algorithm as default task but, if required, different application oriented tasks (e. g. find objects, collect items, etc.) can be assigned to it through a distributed and robust allocation mechanism described below. We integrate these behaviors in a high level robot control architecture implemented with agent based technology and presented in section *C.*

B. Task Allocation System

In order to assign tasks to robots in an effective way, a distributed task allocation mechanism is defined. One of the goals of the defined methods is to be able to adapt both environment changes (e. g. addition of new environment areas) and robots' team changes (e. g. robot failures). Finally the allocation mechanism has to be robust enough for working in critical scenarios like rescuing where broadcast communication may be not available. In the multi-robot system a central server is considered in order to interoperate with external software applications and to provide services to robots. The main phases of the auction algorithm can be summarized as follows:

1. The central server acquires tasks to be executed by the robot team, new tasks can emerge as results of user requests or automatically generated by a reasoning system.

2. The central server analyzes the task definitions and decides how many robots should be assigned to each specific task but do not explicit identify them.

3. Then the robots should be informed about the new tasks to be assigned and the number of robots required.

4. The assignment is the result of a negotiation that each robot leads by itself.

These last two points have to be investigated in order to make the proposed task allocation system adaptive and robust.

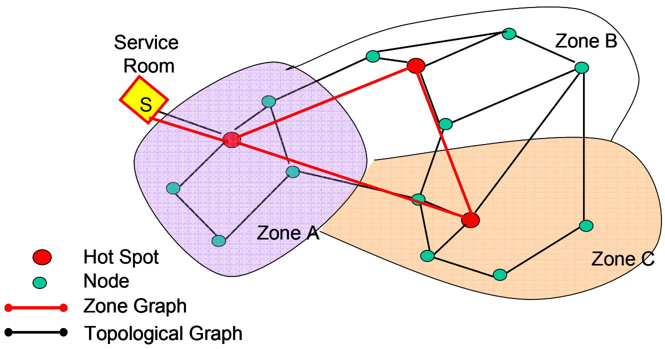

Figure 1. – The graph based model of the environment.

As previously mentioned, the environment is modeled through a graph. For task allocation purpose, it is also divided in zones and for each zone an *hot spot* is defined. The *hot spot* is a local memory (e. g. a cluster of RFID tags - in order to record a larger amount of data) containing information relevant for that zone: blackboard for robots indirect communication (e. g. task allocation messages), robots log (each robot logs in/out entering/leaving the zone), additional relevant task oriented information. At least a Service Room is foreseen within the environment: this is a special zone in which robots can recharge their batteries and the communication with the server system is ensured.

Each application oriented task consists in performing some specific actions in a defined area; it is modeled by the following properties: Priority, Deadline, Required capabilities, and Revenue credits. The priority is critical for scheduling the task while the deadline defines the time within the task has to be completed. In order to effectively perform each task specific robot capabilities are required: this is a relevant point when considering a team of heterogeneous robots.

Once a task is concluded, it determinates an amount of virtual credits that is assigned to the robot as revenue for completing the task; this is a focal aspect of the bidding mechanisms. The "Call For Task", CFT, represents the "call" promoted by the Server System, which play the role of Auctioneer. It mainly contains the task and the auction information, in particular the auction deadline: the time within robots can send a valid bid to acquire the task.

The basic steps of the auctioning process are:

- while performing assigned tasks, robots look for new CFT;
- when a robot gets informed about a new task, it evaluates the acquisition of the new task taking into account: task required capabilities, already assigned tasks, task priority compared with the already scheduled task, and deadline;
- and then the robot decides to start a negotiation process.

The negotiation process is based on a first-price auction:

- each robot, able and interested in performing the task, makes its offer specifying an amount of credits according to its state (task already assigned/performed, battery charge, ..);
- at the expiration of the auction time-out, indicated within the CFT, the task is assigned to the robot that made the highest offer;
- when a robot successfully terminates a task an amount of credit is assigned to it; a Revenue function determines the credit amount considering task revenue credits, priority, task deadline, and completed time.

As mentioned above, in order to work in real scenarios the auction mechanism could not rely completely on the robot communication infrastructure. Two scenarios are considered: with broadcast communication and with local/indirect communication. In the first case the Server System manages and drives the auction with direct server-robots communication that is used for publishing the CFT and collecting the robot offers, this scenario is the most effective considering performance and scalability factors. In the latter scenario, we assume that no Wi-Fi communication is possible and only indirect communication through RFID is available, a mechanism for delivering the CFT to the robots deployed within the environment is required. In order to solve this problem, the proposed idea is to exploit the robots as message couriers and the zone hotspots as mailboxes. The couriers collect auction data from the server, within the service room, and deliver CFT to the hotspots. All the robots can play the role of message courier: they can get CFT information from the server when they go to the Service Room for recharging their battery and then they became able to deliver the CFT to the target hot spots while coming back to the working zones. Hence, in the messenger scenario the server communicates to robots in the service room the new CFTs along with the hotspots where to deliver them and these robots deliver the CFTs to hotspots playing the role of messengers. Once the CFTs are published into the hotspots, robots periodically visit the hotspot of the area in which they are working. Periodically means that they visit the hotspot when they terminate the execution of a task or at fixed interval (polling strategy); the interval may vary in function of the priority of task already assigned to that robot. If a robot is interested in a CFT, it starts the negotiation process by leaving its bid for task on the hotspot blackboard. At the deadline indicated into the CFT, each robot involved in the negotiation process checks the auction result (best offer) in the hotspot, hence the winner auto-allocates the task in its schedule. The Server System will be informed on task allocation (auction results) when robots come back to the service room collecting runtime information. If a CFT doesn't receive offers, the Server System can decide to reschedule the involved task in a new one.

C. Robot Control Architecture

The robot control architecture is described in the following schema. The proposal is based on the result achieved in the Roboswarm project [2], funded by European Community in the Sixth Framework Programme.

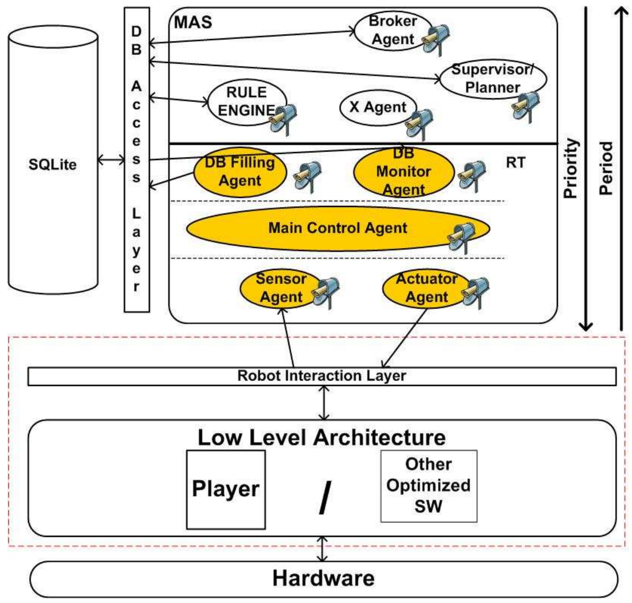

Figure 2. – Robot Control Architecture

We adopt the agent paradigm for modeling the single robot control architecture and the whole multi-robot systems. The architecture involves a dedicated multi-agent system; different kinds of agents can be defined in order to solve/execute specific problems/actions. A reactive agent, Main Control Agent works as the central control process and it can activate reactive behaviors. It executes the real time coordination algorithm and schedule the application oriented task if requested. We introduce two other reactive agents: Sensor Agent and Actuator Agent. The Sensor Agent is in charge of reading input data from sensors and writes them in a shared buffer. Actuator Agent reads robot commands and passes them to robots' hardware. Reactive agents have to be scheduled as high priority tasks in order to guarantee predictable behavior and safety in performing motion.

The Broker Agent negotiates with other robots and, according with the Supervisor (Planner) Agent, models robot tasks that the Main Control Agent will perform. Its behaviors implement the bidding mechanisms. The Rule Engine performs elaboration on robot knowledge [17], dealing with learning activities out of the scope of this paper.

All the agents, implemented as an AgentService solution, could be deployed on board (a dedicated mobile device in our tests), or some of them, the upper level agents (Broker and Planner), can be distributed on server side. Following the AgentService scheduling model, a correspondence agent-thread is adopted within mobile device while a more solutions are available on server side.

A dedicated agent communication channel (ACC) is defined in order to make agents, and other on-board software artifacts interact. Moreover, the ACC manages the inter-robot communication using open standard technologies for sending data to other robots. Hence an agent, deployed inside the robot, is able to send asynchronous messages to other agents with no care about the place where receivers are deployed: the ACC performs the right action for us. According to the operational scenario, the communication service could be degraded in indirect robot communication through RFID tags when WiFi communication is absent.

The agent-robot interaction is modeled through a programming interface which invokes a specific low level Application Programming Interface (API) modeled by the Robot Interaction Layer. The Robot Interaction Layer abstracts the interaction with the software components which act on the hardware. By means of this level of abstraction, agent behaviors are not strongly coupled with the hardware level increasing their reuse and portability. Even this software level is implemented with AgentService [14]. The Robot Interaction Layer works in two ways: emulating the robots behaviors in a virtual 2D environment, directly sending commands and receiving sensors data from the robots. Considering the last mechanism, a couple of IRobot Roomba platforms equipped with RFID readers and Bluetooth adapters are used for real experiments while an instance of AgentService platform has been deployed on a mobile device based on Windows Mobile and .NET Compact Framework. Hence, the system can be configured to work both in simulation and in real robot hardware context in a transparent way from the point of view of client applications.

D. Multi-Robot System Architecture

Considering inter-robots communication, in the scenario in which the WiFi infrastructure is available, it is important to follow software standards in order to be able to interoperate with different system/technology and to improve system maintenance making the Multi Robot System open. The concepts introduced with the Service Oriented Architecture [16] along with the standardization of XML Web Service (SOAP, WSDL, UDDI) lead to the adoption of these standards.

For inter-robot communication, the proposal is that the software agents inside the robot can send data to other robots (agent deployed on) or to the system through the ACC and the Service Access Layer.

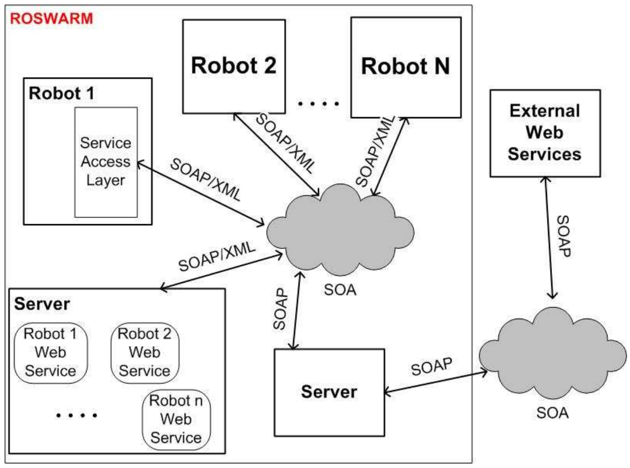

Figure 3. – Robot Layered Architecture

In addition, each robot may promote its activities through a dedicated Web Service (Robot Web Service) placed on the highest and external layer of the robot architecture (server-side).The Multi-Robot Server System can provide services to robots and should be able to orchestrate external services too. In this way everything is a service or is wrapped by a service. Communication between the Robot Web Service and the onboard software layer is based on standard SOAP messages ensuring easy integration with different software system and, if required, semantic expression of the communication acts.

III. Final Condiderations And Conclusions

This paper presents a coordination system for swarm of robots based on agent technology. As demonstrated in literature, the exploitation of RFIDs for getting context aware information system is providing good results. Two approaches for making teams of robots work cooperative are considered. The objective of task allocation is to find the best distribution of the tasks and sub-tasks among the robots belonging to a team. As previously discussed, this problem is widely applied in different application context and the researchers in the field of multi-robot system coordination exploits the works previously done in the field of multi-agent systems, as we have seen for the market-based approaches. On the other hand control coordination algorithms try to make robots behaviors cooperative by directly controlling the actions of each robots;

this approach is closely related to specific tasks the team of robots have to solve. We can state that task allocation operates at an higher level than coordination control algorithms: the former improves the teams' work by assigning, and in some cases dynamically re-allocating, tasks to robots and the latter coordinates robots' actions while they are executing the assigned task. Hence, it seems clear that the two coordination techniques are complementary aspects of the same coordination process and can be applied together in order to improve the effectiveness of the multi-robot systems behaviors.

The proposed coordination strategy is based on context aware information deployed within RFID tags distributed in the environment and does not rely on an internal robot representation of the system. This information infrastructure is used to drive the control behavior of each robot and to distribute tasks among the team. Furthermore, the proposed system is robust against environment changes and do not require direct robot-robot communication: the model of the environment is distributed with the tags and robots can communicate using the tags as mailboxes. The system has been implemented through an ad hoc agent control architecture which provides modularity and scalability. It has been extensively tested in the AgentService environment both in simulation and on real robots. The test case scenarios involve two main phases: team navigation and coverage of the environment with indirect coordination exploiting RFID context information; task composition and allocation based on the distributed auction mechanisms. The tests state the feasibility of the agent architecture and the effectiveness of the integrated coordination approach. The multi-robot architecture exploits the abstraction and the cooperative aspects of the agent paradigm. It can be noticed the adoption of a communication channel transparent for agents which can exchange messages with peers with no care on where receivers are deployed. A service oriented approach is proposed for exposing and promoting the robot team services and easily integrating the system with external software components.

REFERENCES

[1] Myungsik Kim, Nak Young Chonga, RFID-based mobile robot guidance to a stationary target, Mechatronics, Volume 17, Issues 4-5, May-June 2007, Pages 217-229.

[2] Knowledge Environment for Interacting ROBOt SWARMs project, a European project supported within the IST priority of the Sixth Framework Programme for Research and Technological Development. Contract Number: 045255 (www.roboswarm.eu).

[3] M. A. Batalin and G. S. Sukhatme, The analysis of an efficient algorithm for robot coverage and exploration based on sensor network deployment, Proc. IEEE Int. Conf. Robot. Autom., 2005.

[4] V.A. Ziparo, A. Kleiner, B. Nebel, and D. Nardi. RFID-Based Exploration for Large Robot Teams, Proc. of the IEEE Int. Conf. on Robotics and Automation, 2007, pp. 4606-4613.

[5] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, Mapping and localization with RFID technology, Proc. IEEE Int. Conf on Robotics and Automation, vol. 1, 2004.

[6] J. Bohn and F. Mattern, Super-distributed rfid tag infrastructures, Proc. of the 2nd European Symposium on Ambient Intelligence, 2004.

[7] A. Milella, P. Vanadia, G. Cicirelli, and A. Distante, RFID-Based Environment Mapping for Autonomous Mobile Robot Applications, International conference on Advanced intelligent mechatronics, 2007.

[8] G. Roussos, D. Papadogkonas, J. Taylor, D. Airantzis, M. Levene, and M. Zoumboulakis, Shared Memories: A Trail-based Coordination Server for Robot Teams, First International Conference on Robot Communication and Coordination, Athens, Greece, 2007.

[9] J. Werfel and R. Nagpal, Extended Stigmergy in Collective Construction, IEEE Intelligent Systems 21(2), 2006.

[10] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, Market-based multirobot coordination: A survey and analysis. Proc. of the IEEE – Special Issue on Multi-Robot Coordination, 2006.

[11] S. C.Botelho, and R. Alami, M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement, Proc. of the IEEE Intl. Conf. on Robotics and Automation, Michigan, 1999.

[12] M. B., Dias and A. Stentz, Opportunistic Optimization for Market-Based Multirobot Control. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2002, 2002.

[13] R. Zlot and A. Stentz, Complex Task Allocation For Multiple Robots, Proc. of IEEE Int. Conf. on Robotics and Automation, Barcellona, Spain, 2005.

[14] C. Vecchiola, A. Grosso, A. Boccalatte; AgentService: a framework to develop distributed multi-agent systems, Int. J. Agent-Oriented Software Engineering, Vol. 2, No.3 pp. 290 – 323, 2008.

[15] M. Baglietto, G. Cannata, F. Capezio, A. Grosso, A. Sgorbissa, R. Zaccaria, PatrolGRAPH: a Distributed Algorithm for Multi-Robot Patrolling, The 10th International Conference on Intelligent Autonomous Systems, Germany, 2008.

[16] T. Erl, SOA Principles of Service Design, The Prentice Hall Service-Oriented Computing Series from Thomas Erl, 2007.

[17] Vain, J.; Tammet, T.; Kuusik, A.; Juurik, S. (2008). Towards scalable proofs of robot swarm dependability. In: BEC 2008 : 2008 International Biennial Baltic Electronics Conference : Proceedings: 11th Biennial Baltic Electronics Conference, Tallinn University of Technology, October 6-8, 2008, Tallinn, Estonia. IEEE copyright, 2008, pp. 199 - 202.

[18] M. Mamei, F.Zambonelli, Pervasive Pheromone-Based Interaction with RFID Tags, ACM Transactions on Autonomous and Adaptive Systems, 2(2):1-28, ACM Press, 2007.

[19] B. Gerkey, R. T. Vaughan, and A. Howard, The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems, Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), June 2003, 317-323.

[20] M. Baglietto, G. Cannata, F. Capezio, A. Grosso, R. Zaccaria, A Multi-Agent Control Architecture For Large Scale Multirobot System, 27th IASTED Modelling, Identification, and Control (MIC) 2008, Innsbruck, Febbraio 2008.