A New Architecture For Multi-Robot Teams In Market-Based Applications

Uğur GÜREL [1], Osman PARLAKTUNA [2]

[1]Eskişehir Osmangazi Üniversitesi, Bilgisayar Mühendisliği, Eskişehir, Türkiye
ugurel@ogu.edu.tr
[2]Eskişehir Osmangazi Üniversitesi, Elektrik Elektronik Mühendisliği, Eskişehir, Türkiye
oparlak@ ogu.edu.tr

Abstract

This paper describes architecture for multi agent mobile robot systems in market-based task allocation systems. Constructing architecture for multi-agent mobile robots system is heavily studied in the literature. Also market-based task allocation has been taken considerable attention in the literature. But combining these two areas is a new subject for the researchers. In this paper, we developed architecture for market-based task allocation systems and applied to an example with three mobile robots. In the study, a mobile robot contains 5 agents; each of these agents has a unique responsibility in the system. In the study agents communicate with each other through Open Agent Architecture (OAA). Also simulations on MobileSim simulator program are conducted to show the effectiveness of the proposed architecture.

.

1. Introduction

In mobile robot domain, many applications require robustness, efficiency and more tasks to be done in parallel. In order to accomplish these requirements first of all a proper and flexible architecture has to be constructed. Agent-based architectures are quite preferable for these kinds of architectures. Dividing each sub task to a software agent increases robustness because if a software agent fails it does not affect whole system. Only this failed agent's ability disappears in the system. But the system still works with the remaining agents.

In the literature multi-robot task allocation (MRTA) has taken significant attention. While solving the task allocation problem there are various methodologies. In MRTA problems centralized ([1][2]), distributed [5], and hybrid ([3][4])approaches are addressed in the literature. In centralized approaches all the plans are made by a single robot called planner. In the centralized approach all data is gathered by the planner and an optimal plan can be done. But since it is done only by one robot, any failure of the planner causes the whole team to fail.

In the distributed approach every team member of the robot team is responsible of its own plan generation based on its local information and states [6]. There are several multi robot architectures for MRTA problem in the distributed approach. ALLIENCE [7] is one of the leading papers in the literature. Another approach is developed by Sandholm [8] which is based on contract net protocol (CNP). In the study market-based negotiation is applied based on contract net protocol rules.

Gerkey and Mataric [9] has used a publish/subscribe communication methodology for their task allocation via distributed negotiation problem. In the study robots' capabilities are also considered. Sarıel [10] has developed an architecture named Distributed and Efficient Multi-Robot - Cooperation Framework (DEMiR-CF). In her study architecture is developed for multi robot teams to achieve global missions. This framework is based on distributed market approach.

In recent years market-based approaches become more and more popular. In the market-based approach each robot member builds its own plan using its local information. But besides this, mobile robot can participate in auctions to take unassigned tasks in the market. Using market- based approaches; a user can combine both centralized and distributed systems advantages. In the literature Stentz and Dias [11] first introduced the concept of using a market approach to coordinate multiple robots to cooperatively complete a task.

The rest of the paper is organized as follows. In section 2, the proposed architecture is going to be explained. In section 3, application of the proposed method and experimental work is going to be presented. In the final section conclusions and the feature work will be given.

2. Proposed Architecture

In the proposed architecture all of the communications between agents are made via Open Agent Architecture (OAA) structure (Figure 1). Open Agent Architecture is a framework for integrating a community of heterogeneous software agents in a distributed environment. SRI's Open Agent Architecture (OAA), a research framework for constructing agent-based systems, makes it possible for software services to be provided through the cooperative efforts of distributed collections of autonomous agents.[12] In the OAA messaging is made through ICL messages. The ICL messages contain the sender, the receiver and the message information in the message body. All of the messages are flow through the facilitator agent. The Facilitator agent is a pre constructed agent in the OAA environment.
ICL message can be constructed as follows:

```
ICLTerm *list=icl_NewList(NULL);
ICLTerm *RX=icl_NewInt(rx);
icl_AddToList(list,icl_NewInt(rx),TRUE);
goal = icl_NewStruct("var_name", 1, list);
```

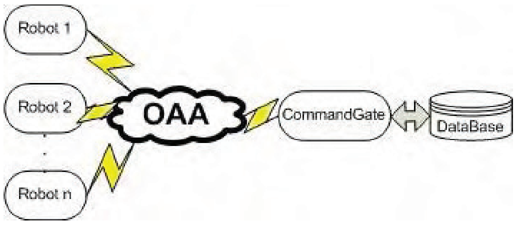

Fig. 1. Communication Architecture via OAA

In the proposed architecture there are six different agents. These agents named as CommandGate, GetWork, InsertWork, Auction, Action and MobileRobot. Only MobileRobot agent is a physical agent other five agents are software agents. Some of the agents are exists in each robot but CommandGate is unique in the system. The agents in a robot are shown in figure 2.

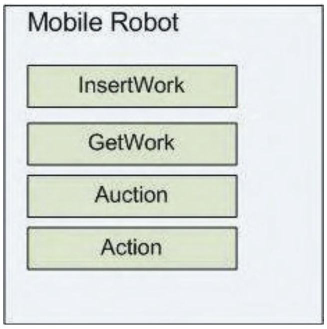

Fig. 2. Agents Present in a Mobile Robot.

Agents are explained more details below:

InsertWork : This agent is responsible for produce service or task for the other robots. If a new job is produced its requirements, position is sent to CommandGate over Open Agent Architecture (OAA) with a special sentence. In the application tasks are produced in a random time period between 30 and 600 seconds.

Yeniw, X_Poz_of_the_Task, Y_Poz_of_the_Task

When CommandGate receives this special sentence from *InsertWork,* it saves this information to a special data file called database. After sending this sentence to *CommandGate InsertWork* does not wait any response from *CommandGate*. Only if there is an error at *CommandGate* agent error code is send back to *InsertWork*.

GetWork **:** This agent is responsible for gathering unscheduled task from database via *CommandGate* with a special sentence

Yeniw, Robot_Id

When this command is received by *CommandGate*, it sends back the tasks information to the robot who send Yeniw command. The sutructure of the answer given below.

Yeniw, Robot_Id,Task_Id X_Poz_of_the_Task, Y_Poz_of_the_Task

If there is no unallocated task an error code is send back to the robot.

Auction *:* This Agent is responsible for managing auction process that will be used for task allocation. .With this agent mobile robot can open an auction and receive bid from other robots or bid to an opened auction process. All of the auction process data is stored in the database file. To store bid value the following sentence is used

Auc, Robot_Id, Task,Cost

When this sentence is received by *CommandGate* the values are parsed and stored in database file.

Action: This agent is responsible for mobile robot movements. It stores current position a gives necessary commands to achieve taken tasks.

Structure of the agents is shown in Figure 3.

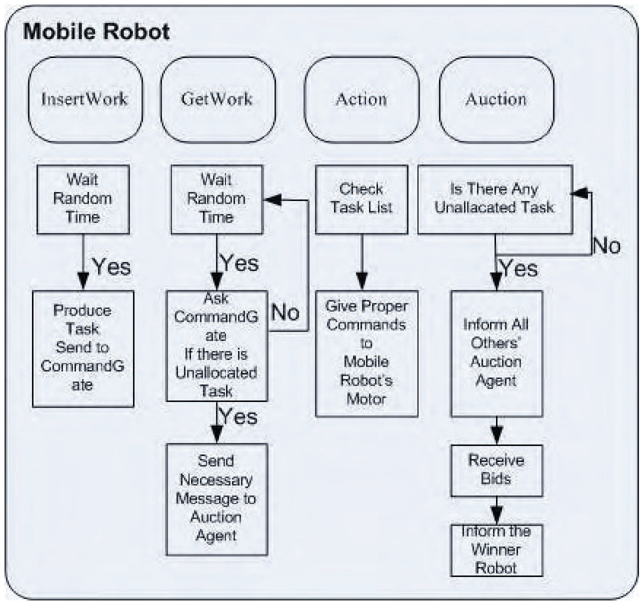

Fig. 2. Structure of the agents

Database File: This file is a special file which stores important information about the agents. To avoid multiple access at the same time only *CommandGate* has access right to this file. Other agents query or store data via *CommandGate* with special commands. In the database tasks, auction data is stored.

3. Aplication

In market based applications there are two main roles. The first one is the auctioneer, and the second one is bidder. The auctioneer offers a task to other robots and robots response to this offer to take the offered task. Here between the auctioneer and the bidder there should be a meaningful exchange of an item. The most valuable item for the robot is its energy because without energy mobile robot can not do any thing. Therefore, this energy is the most suitable exchange item in market-based applications. Also the energy consumed by the robot is directly proportional to the distance that the robot travels. Under these conditions while calculating the cost of the task the distance traveled by the robot is going to be used. The shortest path is determined by the Dijkstra's Shortest Path Algorithm.

In the study each robot (MR_i where i=1,2, …n) has its own agents as explained before and in the market each robot has two basic roles; auctioneer (Auc_i ,where i=1,2, …n), and the bidder (Bid_i ,where i=1,2, …n). With the auctioneer role mobile robot holds auctions for unallocated tasks and with the bidder role they submit a proposal for the unallocated tasks.

In the market, tasks are produced by the *InsertWork* agents that are present in each mobile robot. At pre-defined time internals if there is any unallocated task *Auction* agent announces the task and waits for a pre-defined time interval for receiving bids from other robots. When the time is up, the Auction agent determines the winner and announces it. While determining the winner, the bid that has the lowest value is chosen. Because the lowest value also guarantees the shortest path is going to be chosen which is better for the robot and the mobile robot group. If there is no bid or there is no winner, the task is written back to Database file for future auctions.

The flowchart of the auction is described in Fig.4.

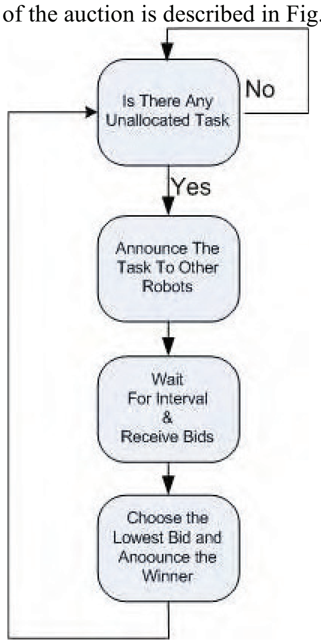

Fig. 4. The Flowchart of the Auction

In the application the only task is visiting certain places. There are three homogeneous robots in the system. The robot is assumed to be successful if it visits the node at which the task is defined.

The simulation environment is given in figure 5.

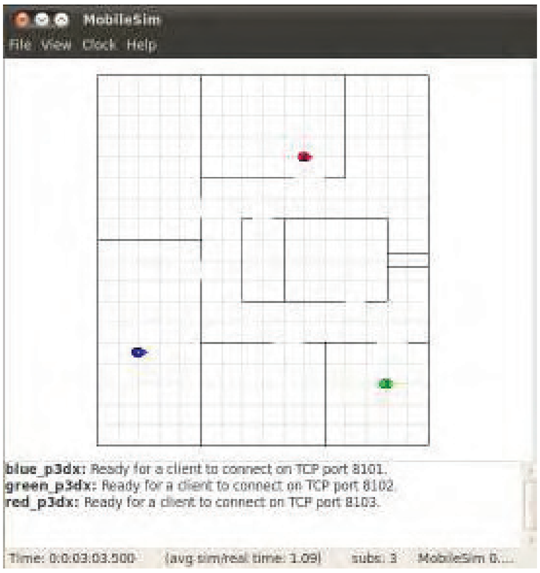

Fig. 5. Simulation Environment

For example there is new visit task at point (6000,8500) and robots current positions stated in table 1 according to this auction Robot 1 has the lowest bid so the auctioneer give the task to Robot 1.

Table 1. Auction

Task at point (6000,8500)			
Robot	Current Pozition	Bid	Status
Robot 1(Blue)	2000,4500	7656,85	Lowest
Robot 2(Green)	14000,3000	13500	
Robot 3(Red)	10000,14000	1059	

This process continues until there is no unallocated task is present.

4. Conclusions and Future Work

In this paper, a new agent-based architecture for market-based systems is proposed. Communication framework is handled by Open Agent Architecture. The proposed architecture is tested on MobileSim Simulator program. The initial results prove that the proposed architecture is suitable and working reasonably for multi-agent systems in market-based applications.

We plan to extend this study for the dynamically changing tasks. Also as a feature work we plan to increase variety of the tasks and applying it to heterogeneous mobile robot teams. Another future work would be application of this approach to real world robot applications.

5. References

[1] B Brumitt and A. Stenz. “GRAMMPS: A generalized mission planner for multiple mobile robots”. *In Proceedings of the IEEE International Conference on Robotics and Automation*, 1998

[2] P. Caloud, W. Choi, J. Latombe, C. Le Pape, and M. Yim. “Indoor automation with many mobile robots”. *In Proceedings of the IEEE International Workshop on Intelligent Robotics and Systems (IROS)*, 1990.

[3] M. B. Dias and A. Stenz. “Opportunistic optimization for market-based multirobot contro”l. *In Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2714–2720, Lausanne, Switzwerland, 2002.

[4] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. ”A practical decision-theoretic approach to multi-robot mapping and exploration”. *In Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2714–2720, 2003.

[5] E.H Ostergaard,. et.al. “Distributed multi-robot task allocation for emergency handling”, Mataric, M.J.; Sukhatme, G.S.Intelligent Robots and Systems, 2001. *Proceedings. 2001 IEEE/RSJ International Conference on Volume: 2 Digital Object Identifier: 10.1109/IROS.2001.976270* Publication Year: 2001 , Page(s): 821 - 826 vol.2

[6] B. Kaleci, O. Parlaktuna, and M. Ozkan “Market-Based Task Allocation Method Using Exploratory Learning for Heterogeneous Multi-Robot Team”, *International Symposium on INnovations in Intelligent SysTems and Applications* (INISTA 2010), June 2010, Kayseri, Turkey.

[7] L. E. Parker. “ALLIANCE: An architecture for fault-tolerant multi-robot cooperation*”. IEEE Transactions on Robotics and Automation,* 14(2):220–240, 1998.

[8]T. Sandholm.” An implementatin of the contract net protocol based on marginal cost calculations”. *In Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, 1993.

[9] B.P. Gerkey and M.J. Mataric. “Murdoch: Publish/subscribe task allocation for heterogeneous agents”. *In Proceedings of the Fourth International Conference on Autonomous Agents*, pages 203–204, Barcelona, Spain, 2000.

[10] S. Sariel and T.R. Balch, "Efficient Bids on Task Allocation for Multi-Robot Exploration*", in Proc. FLAIRS Conference*, 2006, pp.116-121.

[11] A .Stentz. and M. B Dias,., “A Free Market Architecture for Coordinating Multiple Robots”, *Technical Report, CMU-RI-TR-99-42, Robotics Institute*, Carnegie Mellon University, 1999.

[12] A. Cheyer, and D. Martin, “The Open Agent Architecture”. *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 4 , no. 1, pp. 143-148, March 2001.