Motivation and Context-Based Multi-Robot Architecture for Dynamic Task, Role and Behavior Selections

Dong-Hyun Lee and Jong-Hwan Kim

Department of Electrical Engineering and Computer Science
KAIST
Daejeon, Korea
{dhlee,johkim}@rit.kaist.ac.kr
http://rit.kaist.ac.kr

Abstract. This paper proposes a multi-robot coordination architecture for dynamic task, role and behavior selections. The proposed architecture employs the motivation of task, the utility of role, a probabilistic behavior selection and a team strategy for efficient multi-robot coordination. Multiple robots in a team can coordinate with each other by selecting appropriate task, role and behavior in adversarial and dynamic environment. The effectiveness of the proposed architecture is demonstrated in dynamic environment robot soccer by carrying out computer simulation and real environment.

Keywords: Multi-robot coordination, probabilistic behavior selection, team strategy.

1 Introduction

Many practical applications such as reconnaissance and surveillance can be accomplished more effectively and efficiently by using a team of robots rather than using just a single robot to save both time and effort. There might be more than just one task that need to be accomplished in such applications, where each task can be completed by robots with different roles. Consequently, robots in a team are required to select their task, role and behavior in an appropriate manner. Especially in adversarial and dynamic environment, poor task, role and behavior selections may cause tasks more difficult or even impossible to be completed. The task in this paper requires loosely-coupled coordination such as multi-robot object tracking, reconnaissance and exploration. The robots that perform the same task have a similar level of capability, possibly with different resources. There were behavior-based approaches ([1]-[2]) and market-based approach to deal with such problems ([3]-[6]).

This paper proposes a multi-robot coordination architecture, called MCMRA (Motivation and Context-based Multi-robot Architecture), which deals with task, role and behavior selections issues. The proposed architecture uses the

J.-H. Kim et al. (Eds.): FIRA 2009, LNCS 5744, pp. 161–170, 2009.

motivation of a task, which considers robot's capability, and the required capability for the task to maximize the efficiency of resource usages. To select a task, each robot calculates the motivation of each task, generates a context information which consists of environmental state and other robot's condition, and calculates a task strategy ratio which controls the weight of each task based on the strategy and environmental state. After the task selection, one of the roles that belongs to the selected task is selected based on the priority among roles, current role of other robots and the utility of each role. The confabulation method is used for behavior selection, which considers the robot's task, role and environmental state [7][8]. The problem which the architecture deals with can be categorized as a *ST-SR* (Single-Task robots and Single-Robot tasks) problem, where each robot is capable of executing at most one task at a time and some tasks can require multiple robots, in terms of the taxonomy of multi-robot task allocation [9]. The proposed architecture is applied to the game of robot soccer [10].

The rest of this paper is organized as follows. Section II explains the proposed architecture, MCMRA, in detail. In section III, results are described in detail. Concluding remarks follow in Section IV.

2 MCMRA

Motivation and Context-based Multirobot Architecture (MCMRA) considers a motivation of robot for each task, a context of environment and other robots to select its task, role and behavior properly in a distributed manner. The architecture is shown in Fig. 1.

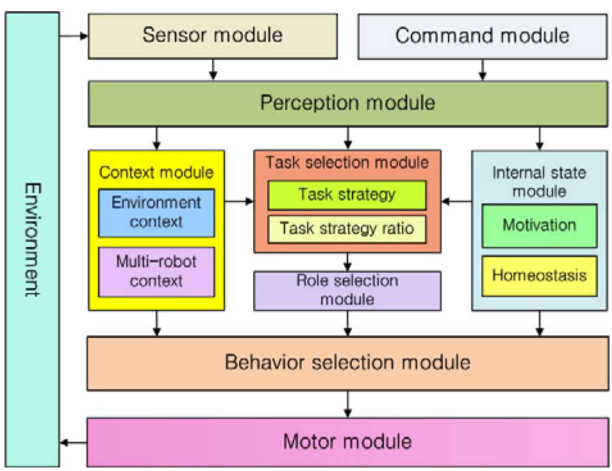


Fig. 1. MCMRA

The sensor module senses the environment and the command module receives commands from user. The perception module generates perception information from sensor data. The internal state module consists of motivation and homeostasis. The context module consists of the environment context and the multi-robot context. The task, role and behavior selection modules are introduced. The Detailed description of each module is provided in the following.

2.1 Internal State Module

The internal state module consists of motivation and homeostasis.

Motivation. The motivation represents a desire to do a task. It is calculated by the capability of the robot and the cost of performing the task. The motivation of robot i $(i = 1, ..., n)$ on the task j $(j = 1, ..., s)$ can be defined as

$$m_i^j = f_m(C_i, P_i, T^j) \tag{1}$$

where C_i is the robot capability in a vector form, which represents the robot's ability such as localization and box pushing. P_i is the perception vector such as the distance from a target and the number of objects around the robot. T^j is the requirement vector of task j and represents what kind of capabilities the task requires and how the cost for the task is estimated. Robot can calculate the degree of its ability on the task from C_i and T^j and the estimated cost on the task from P_i and T^j.

By utilizing motivation, robot can inform the other robots what kind of tasks they can do better considering its capability and the estimated cost for the task. Each robot considers information from the others as well as its own motivation when the task is selected.

Homeostasis. The meaning of homeostasis is the ability or tendency of an organism or cell to maintain internal equilibrium by adjusting its physiological processes. This concept was applied to artificial robots to express the physical condition such as desire to sleep, eat and evacuate [11]. Homeostasis in this paper is used to keep robot in a stable condition when a task is performed. It can be used to check the malfunction of hardware resources such as sensor and actuator. This can increase the robustness of multi-robot coordination.

2.2 Context Module

In distributed multi-robot system, each robot should be able to select its own task and role by itself. For this, the information such as environment and other robot's state should be shared among robots in a team not to compete each other and to perform an efficient coordination. Thus, the environment context module and the multi-robot context module are provided for information sharing.

Environment Context Module. The environment context module combines the local information from the robot's perception module and the other robot's perception from the multi-robot context to estimate the environmental state of robot team. It does not necessarily mean that whole perception data of robots in a team should be shared.

Multi-Robot Context Module. In MCMRA, robots broadcast their motivations on tasks, position and messages such as *Help* and *Abandon* and they record the received information in the multi-robot context. By doing this, all robots in a team can have the identical multi-robot context and each of them can select a task without any conflict by using the multi-robot context and the same task and role selections algorithm. When a robot found a problem with its homeostasis, it sends *Abandon* message to others. *Help* message is also used when the robot requires a help from other robots.

2.3 Task Selection Module

Task selection module selects a task based on the internal state of the robot, task strategy, task strategy ratio of the team, the multi-robot context and messages from other robots.

Task Strategy and Task Strategy Ratio. The task strategy is a set of weights of tasks. The more weights a task gets, the more robots can select the task. The weights in a task strategy are predefined by user. In case of robot soccer, there may be two tasks, offense and defense and the user can make offensive strategy in which the weight of the offense task is selected to be higher than the one of defense task. The *l-th* task strategy, str_l is defined as follows:

$$str_l = [w_1, w_2, ..., w_s] \ (l = 1, ..., e) \tag{2}$$

$$\sum_{k=1}^{s} w_k = 1 \qquad (0 \leq w_k \leq 1)$$

where w_k is the weight of $task_k$. Total sum of the weights are one so that the weight can express the relative proportion among tasks.

Once the task strategy is decided, all tasks can get weights and then the task strategy ratio adjusts the weights of tasks based on the environmental changes. For example, consider there are two areas A and B in Mars and the team of robots explores the areas to find an ice. The user first selects the *strategy 1* which puts higher weight on area A to make robots to selects area A rather than B. During the exploration, the robot which had explored the area B finds the traces of the river and broadcasts the information to other robots. Then the strategy ratio of all robots in a team adjusts the weights of the tasks to make the weight of the task B higher than that of task A.

By using the strategy, the user can decide the weights of tasks based on the user's knowledge and the robots can efficiently adjust the weights of tasks by using the strategy ratio.

Task Selection. The first step of task selection is to check the homeostasis of robot to decide whether it is available for performing tasks. The second step is to check if there is any robot which requested a help by checking *Help* message from multi-robot context. The final step is to select a task based on the motivation, the number of allocated robots for the task and the other robot's motivations in the multi-robot context.

2.4 Role Selection Module

A task is composed of several roles and each role is performed by a robot to accomplish the task. Each role in a task has different priority based on the relative importance among roles so that the role with higher priority should be selected by robots prior to other roles with lower priority. Robots with the same task should consider each other to select a role without conflict. As we mentioned in Section I, robots with the same task have similar capabilities. Thus, the role selection algorithm does not consider the hardware capability of each robot, instead it considers the utility of each robot on each role. The utility of robot i on role k $(j = 1, ..., v)$ in task j can be defined as

$$u_i^{jk} = f_u(Z_i, V^{jk}) \tag{3}$$

where Z_i is the vector of robot information such as the location of robot and the distance from a target. Z_i can be obtained from the multi-robot context. Note that each robot can calculate the other robot's utility by using the multi-robot context. V^{jk} is the role information vector such as the location or the area that the role covers.

2.5 Behavior Selection Module

The behavior selection module uses confabulation method [7]. Suppose that there are assumed facts, α and β and the conclusion, ϵ. This method assumes that the maximization of cogency $p(\alpha\beta \mid \epsilon)$ is equivalent to the maximized product of $p(\alpha \mid \epsilon) \cdot p(\beta \mid \epsilon)$. That is, if sensor information is used as the assumed fact and robot's behavior is considered as the conclusion, then the behavior that maximizes the probability can be considered as the most suitable behavior of a robot in the situation. This method assumed that all probabilities $p(j \mid l)$ between symbols j and l are known. The confabulation method was proposed for selecting the behavior of software robot [8].

The task, role, internal state and the information from context are used as assumed facts and the robot's behavior as the conclusion. The behavior of robot i on the $role_k$ in $task_j$ is defined as

$$beh_i^{jk} = \arg\max_{\epsilon}(p(\alpha \mid \epsilon) \cdot p(\beta \mid \epsilon) \cdot p(\gamma \mid \epsilon) \cdot p(\delta \mid \epsilon)) \tag{4}$$

where $\alpha, \beta, \gamma, \delta$ and ϵ are the symbolized values of $task_j$, $role_k$, environment context, perception and beh_i^{jk}, respectively.

3 Experiments

The proposed architecture was applied to the robot soccer domain. The robot soccer has two tasks, offense and defense tasks. The offense task has five roles, *Striker*, *Forward*, *CenterWing*, *LeftWing* and *RightWing* and the defense task also has five roles, *Goalkeeper*, *Sweeper*, *CenterBack*, *LeftBack* and *RightBack*.

There are eight behaviors, *Pass*, *Shoot*, *CatchBall*, *Penetration*, *Backup*, *intercept*, *GoRoleArea* and *Wait*. Two kinds of robots were defined, offensive and defensive robots, by their capability. The offensive robot can move faster than defensive one and has lighter and slender frame structure. The defensive robot, on the other hand, moves slower but heavier and wider frame structure can make the robot perform efficient defense. The motivation of robot for each task was calculated by the robot's capability on the task and the distance from the opponent team's goal. The utility of each role was calculated by the robot's location on the playground and the distance from a ball.

Task selection with different task strategy. This simulation demonstrates the task selection of robots in a team with different task strategy and environmental changes.

Three strategies, general, offensive and defensive strategies were defined. The general strategy puts the same weights on both offense and defense task, respectively. The offensive and defensive strategies put more weight on offense and defense task respectively. After the strategy is decided, the weights can be changed by the task strategy ratio which considers environmental changes. The time ratio ($\frac{current\ time}{full\ time}$), the team which got a ball (three conditions of the ball: free ball, home team ball, opponent team ball) and the score ratio ($\frac{Home\ team\ score - Opponent\ team\ score}{Max\ score}$) were considered as the environmental changes in robot soccer domain.

In the simulation, a team consisted of total six robots which consist of three offensive robots and the other three defensive ones. The offensive robot has higher capability on offense task than defense task and the defensive robot has higher

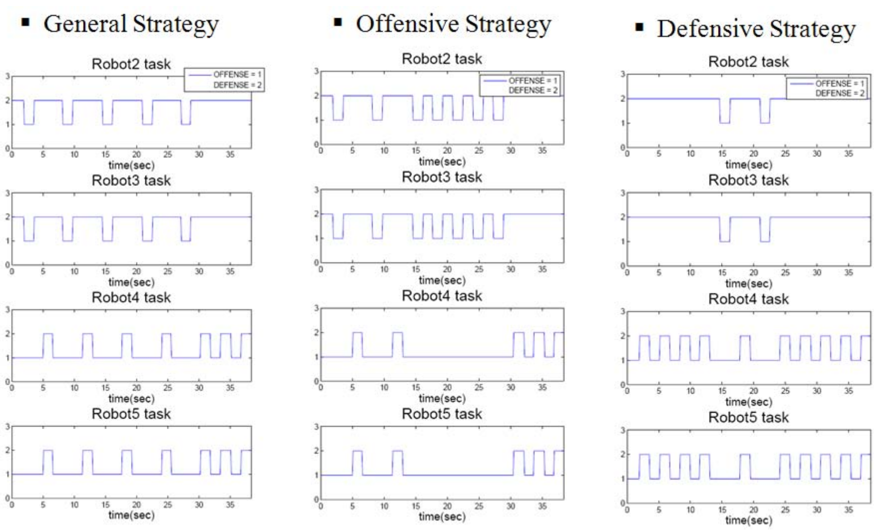
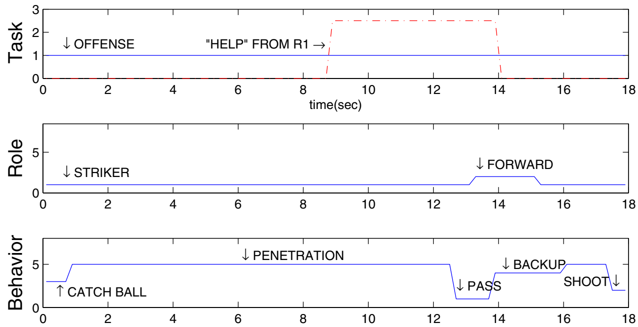
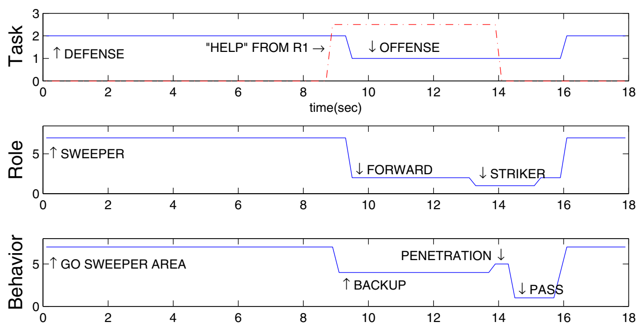


Fig. 2. Task Selection on Different Strategies



(a)



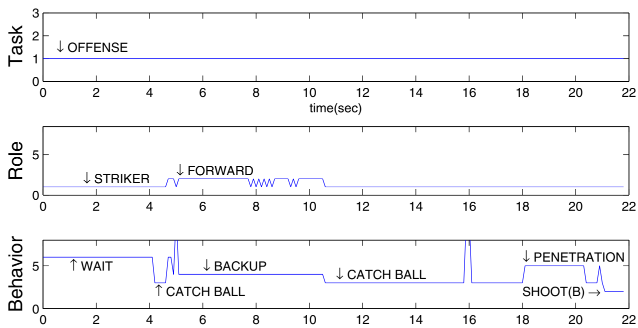
(b)

Fig. 3. Task, Role and Behavior selections. (a) $robot_1^G$, (b) $robot_3^G$.

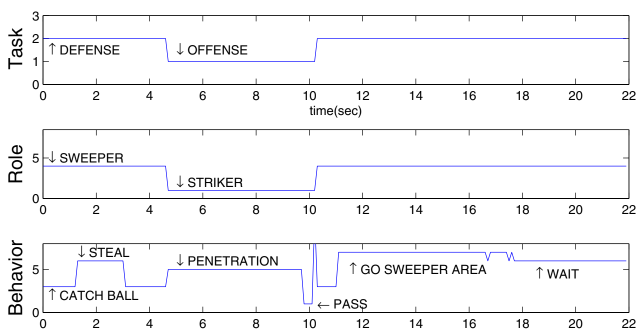
capability on defense task than offense task. The strategy was selected before the simulation. The predefined environmental changes were applied to the team. The time ratio was increased from 0 to 1 until the end of the play and the ball owner was changed like Free ball ⇒ Home team ball ⇒ Free ball ⇒ Opponent team ball. The score ratio was changed; 0:0 ⇒ 0:1 ⇒ 0:2 ⇒ 1:2 ⇒ 2:2 ⇒ 3:2. Fig. 2 shows the task selection of four robots except robot 1 (*Striker*) and robot 6 (*Goalkeeper*) (Two robots do not change their task because both roles are the highest priority roles in each task).

In Fig. 2, the offense task and the defense task were symbolized as 1 and 2, respectively. The robots 2 and 3 were defensive ones and the robot 4 and 5 were offensive ones. The graph shows that the robots 2 and 3 mainly selected defense task and robots 4 and 5 selected offense task. The robots tended to select the offense task around 15 to 25 seconds because the team was losing the game and the time was running out. This tendency occurred regardless of the strategies. It was because the strategy ratio adjusted weights of offense and defense tasks to adapt the environmental change. And the graph also shows that the team selected defense task more than offense task after 35 seconds because they were winning the game and the time was not left much.

Simulation Game with Two Teams. This simulation shows how robots in a team select a task, a role and a behavior dynamically in adversarial and dynamic environment and how they can help each other when a robot requests a help. In the simulation, there were two teams, BLUE and GREEN, and they both had one offensive robot and two defensive robots. Note that the $robot_i$ in BLUE team and the $robot_j$ in GREEN team were represented as $robot_i^B$ and $robot_j^G$,



(a)



(b)

Fig. 4. Task, Role and Behavior Selections. (a) $robot_1^C$, (a) $robot_2^C$

respectively. And the Fig. 3 is the results of the simulation game which show the task, role and behavior selections of GREEN team robots, $robot_1^G$ and $robot_3^G$.

Experiment with Real Platforms. The omniwheel and omnivision-based robot platform, OmniBot was used for the experiment. It can detect landmarks and other robots by using omni mirror and can dribble a ball by omnidirectional mobility. Two teams (CYAN team and MAGENTA team) were competed with each other and each team consisted of one offensive and one defensive robots. The task, role and behavior selections of CYAN team robots are shown in Fig. 4.

The $robot_i$ in CYAN team and the $robot_j$ in MAGENTA team were represented as $robot_i^C$ and $robot_j^M$ respectively. At first, $Striker_1^M$ had the ball but it was intercepted by $Sweeper_3^C$ and it changed its role to $Striker_3^C$. $Sweeper_3^M$ tried to intercept the ball from $Striker_3^C$ and $Striker_3^C$ passed the ball to $Forward_1^C$. After passing the ball, $Striker_3^C$ changed its role to $Sweeper_3^C$ and $Forward_1^C$ changed its role to $Striker_1^C$ and shot the ball to the goal of the MAGENTA team.

4 Conclusions and Future Works

This paper proposed the multi-robot coordination architecture (MCMRA) for dynamic task, role and behavior selections in adversarial and dynamic environment. The proposed architecture, MCMRA, was applied to robot soccer domain. The task strategy ratio was introduced to show that each robot in a team could select its role adaptively based on the team strategy and the environmental changes. Utilizing the task strategy ratio implied that each robot could consider team work as well as its motivation. The simulation game with two teams showed that they could coordinate each other by considering other robot's motivation, requirement for help and environmental changes such as the opponent team robots and the ball possessing team. The experiment with real robots showed that the proposed architecture could be implementable in real robot team.

For the future work, multiple number of robots should be used in a team to test the robustness and scalability of the proposed architecture. In addition, a general framework for creating motivation and utility by considering the task requirement, cost of the task and robot capability such as quality, quantity and energy consumption of each robot should be studied.

References

1. Parker, L.E.: ALLIANCE: An architecture for fault-tolerant multirobot cooperation. IEEE Trans. on Robotics and Automation 14, 220–240 (1998)
2. Werger, B., Mataric, M.J.: Broadcast of Local Eligibility for Multi-Target Observation. In: Proceedings, 5th International Symposium on Distributed Autonomous Robotic Systems, pp. 347–356 (2000)
3. Gerkey, B.P., Mataric, M.J.: Sold!: Auction Methods for Multirobot Coordination. IEEE Trans. on Robotics and Automation 18, 758–768 (2002)

4. Dias, M.B.: Traderbots: A new paradigm for robust and efficient multirobot coordination in dynamic environments, Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh (January 2004)
5. Zlot, R.: An Auction-Based Approach to Complex Task Allocation for Multirobot Teams, Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh (December 2006)
6. Jones, E.G., Browning, B., Bernardine Dias, M., Argall, B., Veloso, M., Stentz, A.: Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In: IEEE International Conference on Robotics and Automation, pp. 570–575 (2006)
7. Hecht-Nielsen, R.: Cogent confabulation. Neural Netw. 18, 111–115 (2005)
8. Kim, J.-H., Cho, S.-H., Kim, Y.-H., Park, I.-W.: Two-layered Confabulation Architecture for an Artificial Creature's Behavior Selection. IEEE Trans. on Systems, Man, and Cybernetics - PART C 38(6), 1–7 (2008)
9. Gerkey, B.P., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multirobot systems. International Journal of Robotics Research 23, 939–954 (2004)
10. Kim, J.-H., Kim, D.-H., Kim, Y.-J., Seow, K.-T.: Soccer Robotics (Springer Tracts in Advanced Robotics), p. 326. Springer, Heidelberg (2004)
11. Kim, J.-H., Lee, K.H., Kim, Y.D.: The Origin of Artificial Species: Genetic Robot. International Journal of Control, Automation, and Systems 3(4), 564–570 (2005)