International Journal of Control and Automation
Vol. 10, No. 2 (2017), pp.173-186
http//dx.doi.org/10.14257/ijca.2017.10.2.15

# Task Management Algorithm for Multi-robot System in Intelligent Space

Jong-Wook Park[1], Joo-Hyung Kim[2] and Dong W. Kim[3*]

[1]*Dept of Electronic Eng., Incheon National University, Incheon, South Korea, 402-752*
[2]*Device Solution Research, Samsung Electronics Corp., 275-18, Samsung 1-ro, Hwaseong-si, Gyeonggi-do, Republic of Korea*
[3]*Dept. of Digital Electronics, Inha Technical College*
*100 Inha-ro, Nam-Gu, Incheon, South Korea, 402-752*
*dwnkim@inhatc.ac.kr*

***Abstract***

*A dynamic and efficient task management algorithm to enable a multi-robot system to effectively handle diverse and unexpected situations is described in this paper. In order to ensure that robots work efficiently in intelligent space, task management of the robots is very important. The proposed task management algorithm effectively manages the tasks of a multi-robot system in intelligent space and schedules the tasks of the robots dynamically using a resource-sharing architecture. It consists of a task scheduling procedure and a task allocation procedure. The task scheduling procedure efficiently schedules the waiting tasks of the multi-robots in the system in order to minimize the total time consumed by all the tasks. The task allocation procedure obtains the working information from the task scheduling procedure to assign jobs to each robot. The performance of the proposed algorithm is evaluated through a simulation program and compared with that of a tree-based task scheduling method. The simulation results show that the waiting time and the task delay to the total time consumed are lower for the proposed method.*

***Keywords**: Task management algorithm, multi-robot system, intelligent space, task scheduling, task allocation*

## 1. Introduction

A network-based robot is a robot connected to a network environment such as the Internet or a local area network (LAN). Such a robot is able to overcome physical or functional limits, which are the disadvantages of conventional stand-alone robots. A network-based robot can be operated remotely and hence provide services over the Internet, unlike stand-alone robots. In recent years, researchers in the field of robotics have been trying to apply network-based robots to new concepts such as intelligent space and Ubiquitous Robotic Companion (URC). Intelligent spaces are rooms or areas that are equipped with sensors, which enable the spaces to perceive and understand what is happening in them. Such a space can support people in terms of providing both information and physical assistance. Consequently, task management of multi-robot systems in intelligent space in order to obtain an appropriate response from the working robots is getting more important. In conventional task management of the robots [3], robot schedules are planned using simple rules and task allocations are fixed. Owing to the rigid architecture, the robot cannot change the task plan and assign tasks dynamically in unexpected situations. Further, if an assigned task is delayed due to unexpected situations, waiting tasks are also delayed till the situation is resolved. Hence, flexible task management algorithms for multi-robot systems need to be researched. This paper

ISSN: 2005-4297 IJCA

presents an efficient task management algorithm for a multi-robot system using a resource sharing architecture (RSA) in intelligent space. It consists of a task scheduling procedure and a task allocation procedure. The task scheduling procedure efficiently schedules the waiting tasks of the multi-robot system in order to minimize the total time consumed by all the tasks. The task allocation procedure obtains the working information from the task scheduling procedure to assign jobs to each robot. The performance of the proposed algorithm is evaluated through a simulation program and compared with that of a tree-based task scheduling procedure. The simulation results show that the waiting time of tasks to be completed by the multi-robot system and the task delay to the total time consumed are lower when using the proposed method.

The remainder of this paper is organized as follows. Section 2 reviews intelligent space. Section 3 describes the proposed task management algorithm for a multi-robot system, including the task scheduling and task allocation procedures. Section 4 shows the simulation results of the proposed algorithm and its comparison with the tree-based method. Section 5 provides a discussion of the results, ideas for further research, and conclusions.

## 2. Intelligent Space

Intelligent spaces are rooms or areas that are equipped with sensors, which enable the spaces to perceive and understand what is happening in them. Such a space can support people in terms of providing both information and physical assistance. Figure 1 is a conceptual figure of an intelligent space. In the space, a human is watched by distributed sensors connected to a network. The space can perceive the human through the sensors and can provide information to the human. The space also controls electrically connected systems for the human. The robots in the space serve as agents that provide physical assistance. The space can be applied to smart home systems. RSA [6] is an architecture used for intelligent spaces. The focus of RSA is to share physical resources and organize them effectively for supporting high-quality services. The structure of RSA consists of physical resources and service objects, as shown in Figure 2. Physical resources are devices such as mobile robots and cameras. Some of them obtain data from the environment and transmit them to service objects via a LAN. The others provide information required by the human. Service objects, such as robot navigation objects and room cleaning service objects, generate useful information from data transmitted by physical resources and provide this information to other services so that an intelligent space can obtain data on the states and abilities of the physical devices connected to the network. Depending on their relation with the other components of the system, service objects are classified into three categories:

1. Fundamental service object: It is a service object directly connected to physical devices. Such objects obtain data from sensors or transmit useful information to the human using devices. Further, the services provide data to inherited service objects.
2. Inherited service object: It provides services organized by more than one object, which may be either fundamental or inherited service objects. It uses the results of fundamental services and generates visual or acoustic information for transmitting data to iSpace service objects.
3. iSpace service object: It is an intelligent service object that collects and processes data from various service objects (*i.e.,* fundamental and inherited service objects) and is able to carry out intelligent tasks such as room cleaning or visitor guidance.

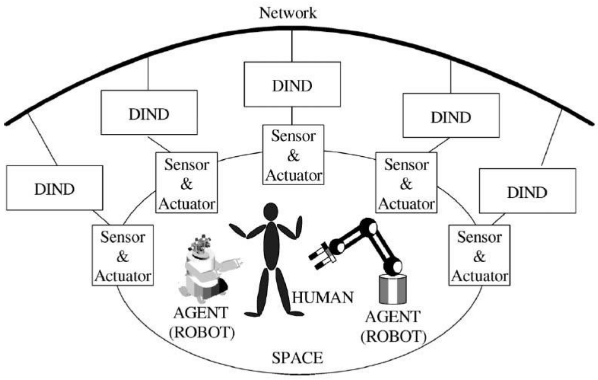



**Figure 1. Conceptual Figure of Intelligent Space**

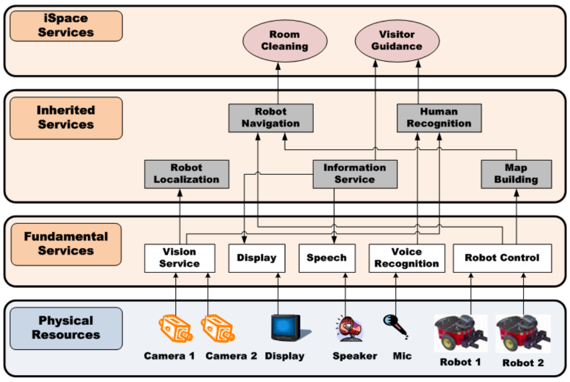



**Figure 2. Resource Sharing Architecture (RSA)**

The RSA framework consists of some robot groups and a task management group. Each robot in a group is controlled by a separate robot agent via certain services. In a task management group, each task management agent determines what the space should do using information obtained from various services.

Robot agent: A robot agent monitors the state of a robot and generates services to execute the assigned tasks. When a task is assigned to a robot agent, the robot starts executing the task using the generated services. If the task being executed is stopped due to some event, the agent conveys the state of the interrupted task to a task management agent so that the task can be continued by other robot agents.

Robot group: Robots can be divided into multiple groups depending on their abilities. In cases where many robots have to cooperate to achieve certain objectives, a task management agent forms a robot group, and adds robots to the group for carrying out the assigned tasks.

Task management agent: A task management agent plans the task schedule of a robot. To manage the tasks, the agent acquires information from certain services and robot agents. Subsequently, it generates a task list for the robots to manage the assigned tasks and allocates the tasks to the best candidates. If interrupted tasks remain in the task list, the agent can reallocate the tasks to other robots. All processes related to task scheduling and task allocation are designed by the task management algorithm, which is incorporated into the agent. We will describe the algorithm for task management in the following section.

Task management group: A task management group is a set of task management agents. If the group coordinates the needed service to an appropriate task management agent, then each task management agent executes smart services.

## 3. Task Management Algorithm for Multi-robot System in Intelligent Space

The proposed task management algorithm consists of two main procedures. One is the task scheduling procedure, which inserts a requested task into the corresponding task list and updates this list. This procedure also plans the schedule of and assigns tasks to each robot. The other is the task allocation procedure, which selects the robot in a robot group that is best suited to an assigned task and performs the task using this robot. The primitive and basic consequence was depicted from [10]. A task management agent selects which robots should carry out an assigned task and informs the robot agent. The responsibilities of the proposed task management algorithm are to delegate an executing task to another robot, to reallocate a delayed task to one or more robots, and to withdraw excess robots from a current task. In this study, we implement two algorithms for the task management of service robots, as described in the subsections.

Figure 3 shows the overall flow of the proposed task-management algorithm. In the following subsections, each procedure depicted will be described in further detail.

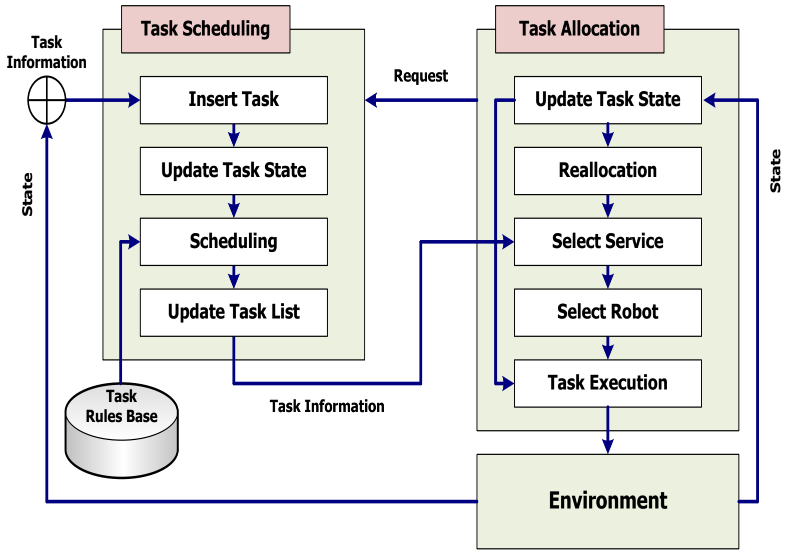



**Figure 3. Overall Flow of Task Management Algorithm**

### 3.1. Task Scheduling Procedure

The role of the task scheduling procedure is to manage tasks that need to be performed and transfer task information to the task allocation procedure. In Figure 4, the overall flowchart of the task scheduling procedure is shown. If a certain situation requires a particular task to be performed, then this task is inserted into the task list. Subsequently, the state of the waiting task is updated. If the task has priority then the task scheduling procedure transfers the corresponding task information to the task allocation procedure. If the task is not a priority task, then the scheduling procedure checks the availability of a robot for the task and plans the task schedule. The proposed task scheduling procedure is termed greedy scheduling.

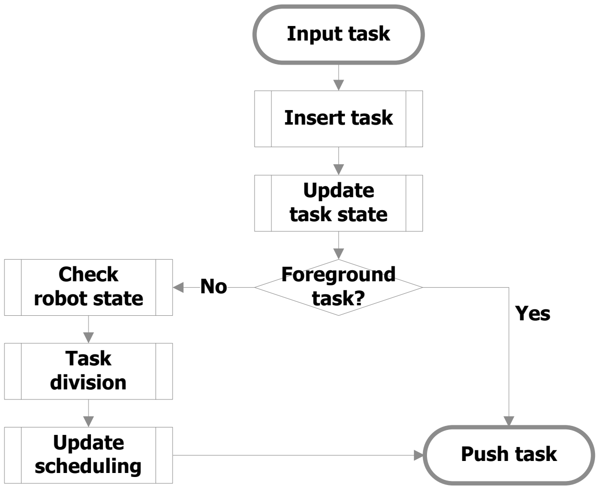



**Figure 4. Flowchart of Task Scheduling Procedure (Greedy Scheduling)**

### 3.2. Task Allocation Procedure

In the task allocation procedure, the state of each robot is considered and the tasks to be performed are assigned to each robot. If the task completion time of a particular robot differs from that in its schedule, the remaining schedule is also adjusted accordingly. The overall flowchart of the task allocation procedure is depicted in Figure 5.

In general, insufficient allocation of physical resources may cause task delay. On the other hand, over-allocation of tasks to a particular robot assuming quick completion of an assigned task is also undesirable. An optimal allocation is that which results in the completion of an assigned task at nearly the scheduled time and the carrying out of as many tasks as possible using the given robots. In the simulation study, we do not consider the number of assigned tasks but the deadline of the assigned tasks for dynamic task reallocation is considered. The objective of balancing allocation is to reallocate a delayed task to other robots and to withdraw excess robots from a current task.

In the case of delayed tasks or tasks completed ahead of time, task management algorithm calculates the reallocation step and estimates the number of robots required for that step. Finally, the task management agent updates the task list of the robots. Next, the robot agent generates the services needed to carry out an assigned task. If a robot agent receives a new task from the task management agent in the middle of an ongoing task, the robot agent stops the current task and reports its state to the task management agent. Then, the agent manages the interrupted tasks and reallocates it to other robot agents.

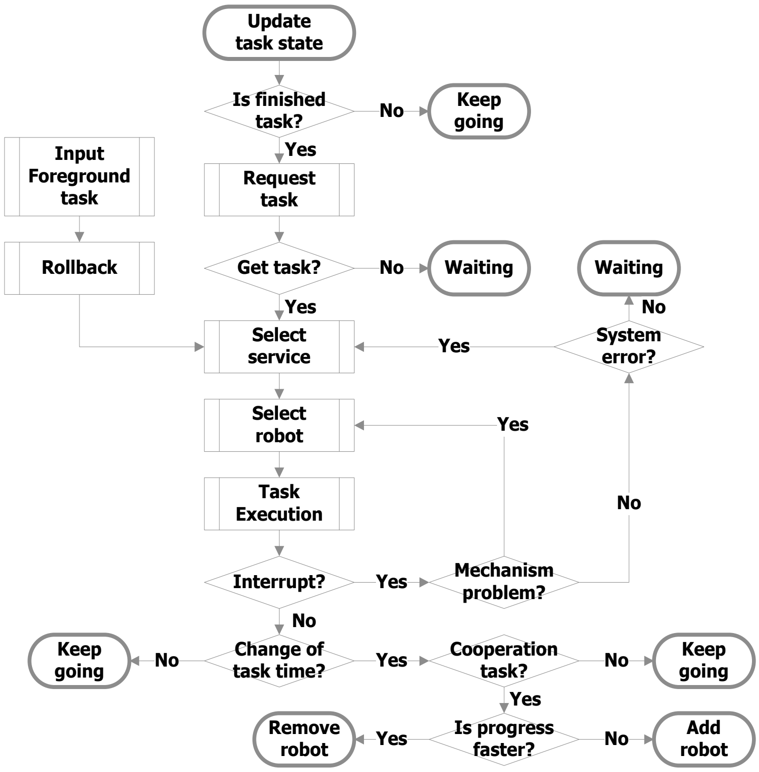



**Figure 5. Flowchart of Task Allocation Procedure**

## 4. Simulation Results

The proposed algorithm was evaluated through three simulation scenarios. In the first scenario, we compare the proposed scheduling procedure with sequential scheduling to evaluate the performance of the task scheduling procedure when there is a change in the task time. In the second scenario, we evaluate the proposed algorithm for task allocation. In the third scenario, we set up a guidance service to evaluate the algorithm assuming that all robots have the same ability.

### 4.1. Simulation Environment

First, we implement a simulator to evaluate the task management algorithm. The simulator is developed using MFC and C/C++ and connected to the architecture described in Section 2. We used the library for mobile robots provided by Active Media Robotics to implement the motion of robots and assume that robots are able to deploy themselves within the environment, avoid obstacles, and plan their path to a destination. It is also assumed that each robot is equipped with a computer to communicate with the agent computer.

Figure 6 shows the simulation program for task scheduling. To evaluate the task scheduling, the number of robots and the time consumed for each task can be set. In the figure, R0 and R1 denote the robots performing the given task. 40 and 30 are task durations (in seconds) consumed for completing the assigned tasks. For example, R0 and R1 take 40 and 30 s, respectively, if sequential scheduling is used; however, in case greedy scheduling is used, R0 and R1 take 30 and 40 s, respectively.

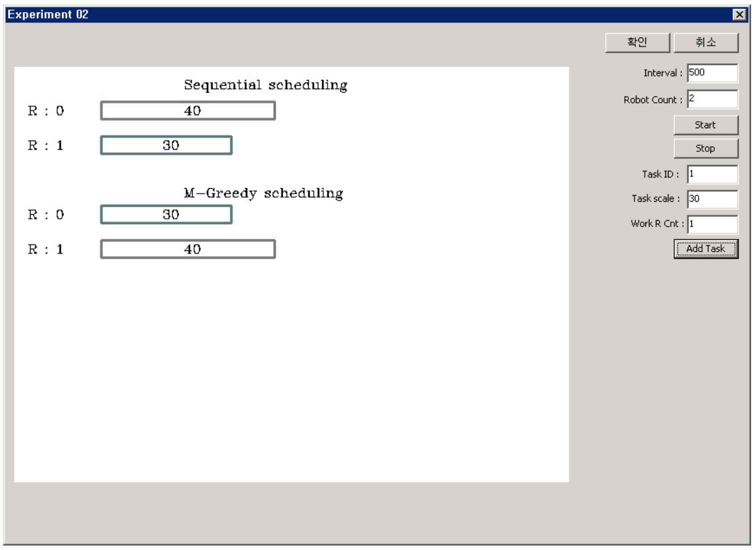



**Figure 6. Simulation Program for Task Scheduling**

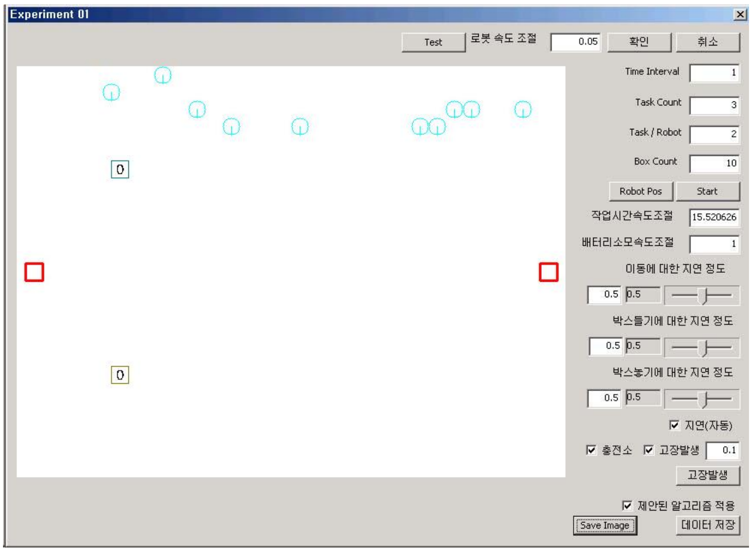



**Figure 7. Simulation Program for Task Allocation**

To evaluate the task allocation procedure, the simulation program shown in Figure 7 is developed. In the program, the number of tasks and robots in a certain space can also be set. The boxes on the left and right side are the battery chargers for the robots. The robots are shown in the top part of the image. The given task in this space is to stack as many small boxes as shown by the “box count” in the figure

### 4.2. Evaluation of Task Scheduling Procedure

In this evaluation, a situation wherein five tasks are to be performed using two robots is considered. The same tasks are completed using two kinds of task scheduling procedures, the sequential and proposed scheduling procedures, although there is a considerable difference between the total time consumed. Time consumed for each task is shown in Table 1. Figure 8 shows that the overall time consumed for the given tasks is different for sequential and proposed scheduling procedures. In Figures. 9-11, Comparison results of total task time, Comparison between ratio of task time of the proposed and sequential scheduling procedures, and Comparison between sum of waiting time of the proposed and sequential scheduling procedures are depicted

**Table 1. Time Consumed For Each Task**

| Table 1. Time consumed for each task | | | | |
|---|---|---|---|---|
| 1st Task | 2nd Task | 3rd Task | 4th Task | 5th Task |
| 50 s | 40 s | 20 s | 40 s | 10 s |

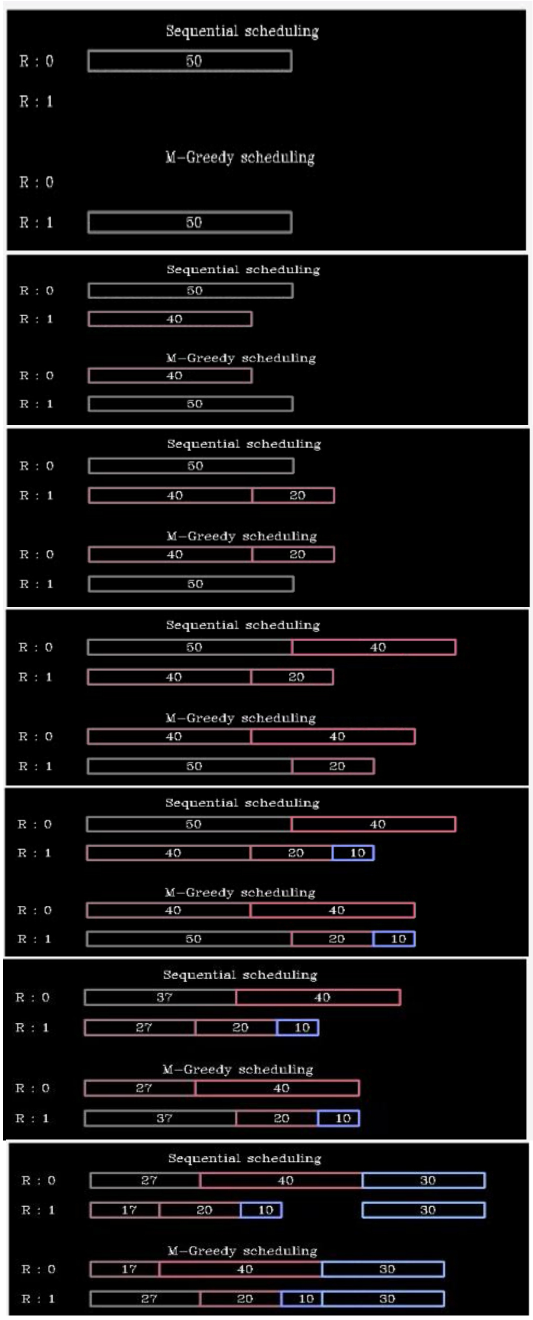



**Figure 8. Comparison between Sequential and Proposed Scheduling Procedures for Five Given Tasks In Terms of Time Consumed for Task Completion**

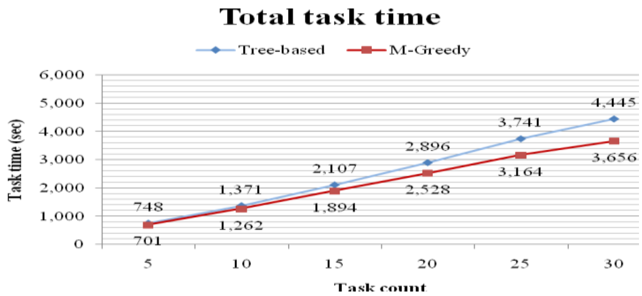



Figure 9. Comparison Results of Total Task Time

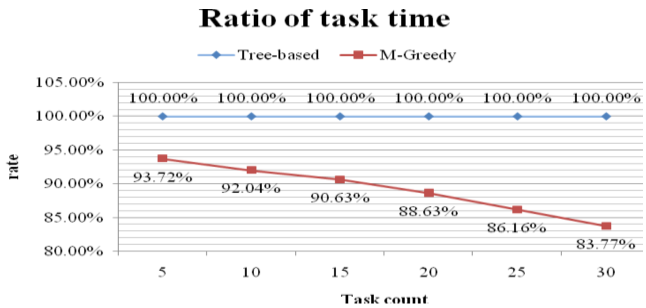



Figure 10. Comparison between Ratio of Task Time of the Proposed and Sequential Scheduling Procedures

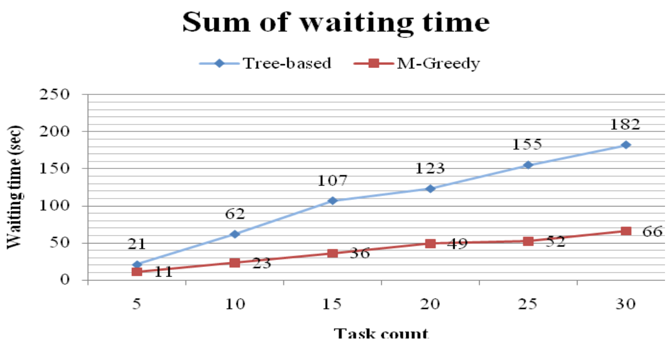



Figure 11. Comparison between Sum of Waiting Time of the Proposed And Sequential Scheduling Procedures

### 4.3. Evaluation of Task Allocation Procedure

In a limited space, there are 10 mobile robots and they have the ability to move something in the space. Using the task allocation procedure, the number of robots working on particular tasks can be adjusted efficiently.. The simulation results from the task allocation procedure are shown in Figures. 12 and 13. In each figure a robot is added or withdrawn to work space. Figure 14 shows the task time comparing with or without prediction.

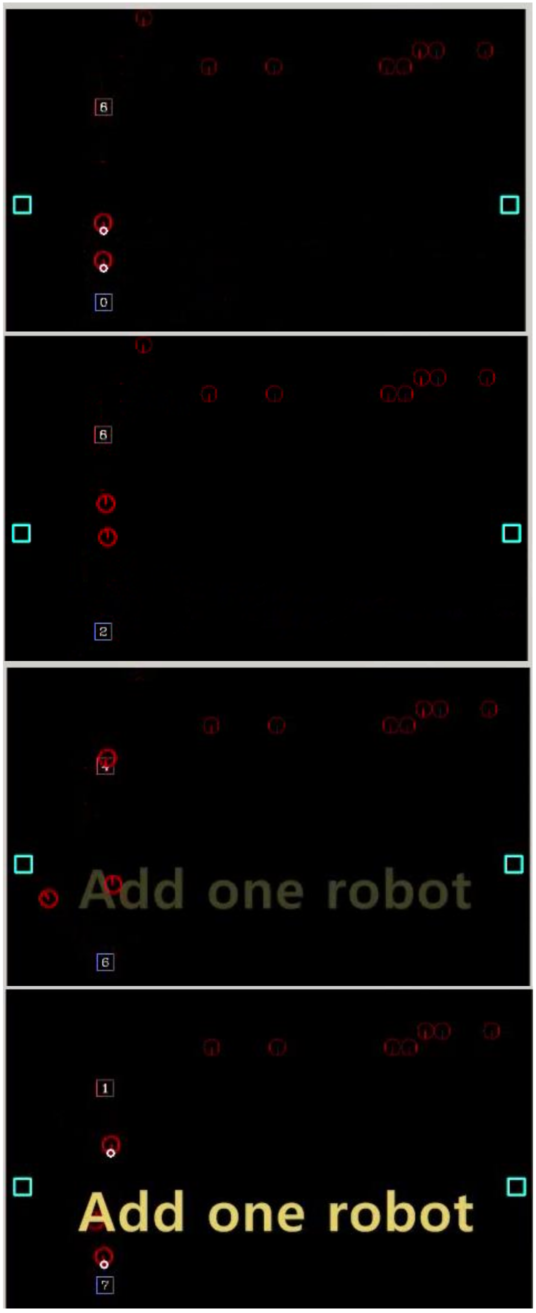



**Figure 12. Simulation Results When a Robot Is Added**

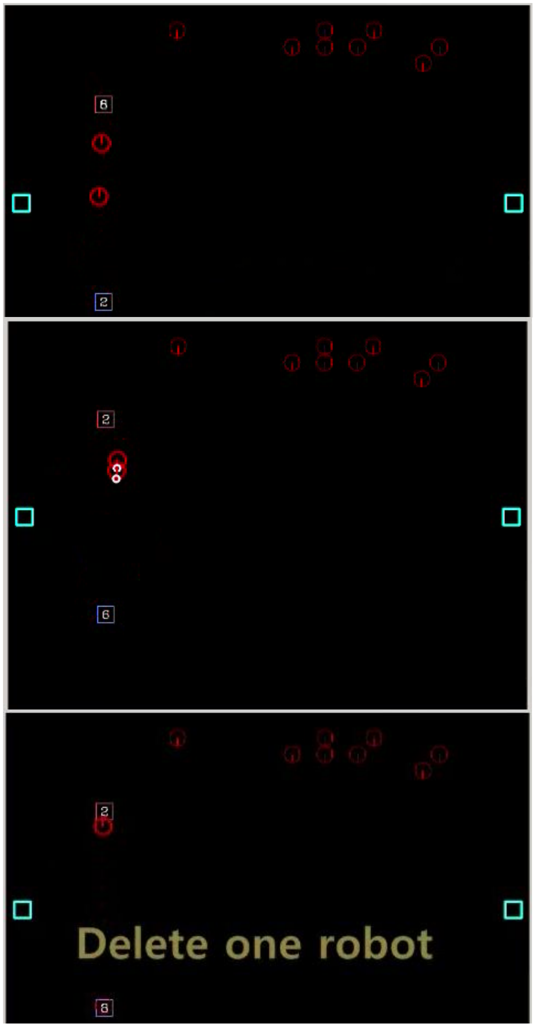



**Figure 13. Simulation Results When a Robot Is Withdrawn**

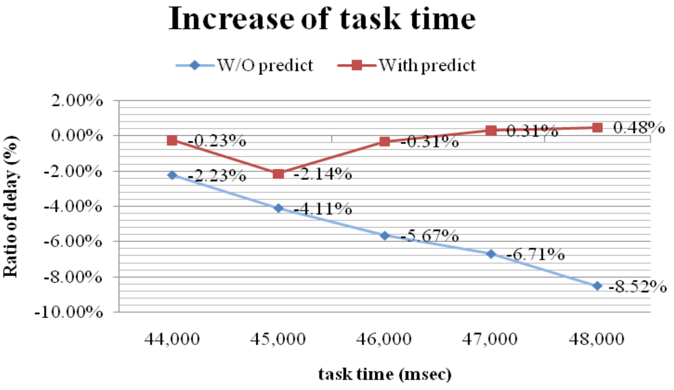



**Figure 14. Task Time With or Without Prediction**

Using this simulation, we verify that the proposed algorithm allocates robots dynamically to complete the assigned task before the deadline.

Guidance Service

The objective of this simulation is to validate the task management algorithm by using an auction-based method to select the most appropriate robot for visitor guidance and the delegation of a task. The scenario for the guidance service is as follows:

Step 1: Three robots are on standby at random positions.

Step 2: A cleaning task is allocated to the three robots.

Step 3: All robots are cleaning in Room 1, 2, and 3.

Step 4: A visitor arrives.

Step 5: An iSpace agent detects the visitor and informs this event to the task management agent, which then selects the most appropriate robot to carry out the guidance service.

Step 6: The selected robot stops the cleaning task and proceeds toward the visitor.

Step 7: The selected robot guides the visitor.

Step 8: Another robot has completed the cleaning task in Room 1 and moves to complete the cleaning task in Room 3.

The results of this simulation are represented in Figure 14, which shows the various sequences in the above scenario in chronological order. In Sequence #1 and Sequence #2, the three red circles are robots that are executing the assigned cleaning task, and the blue circle in Sequence #3 is the robot selected for visitor guidance. Finally, the robot in Room 1 completes its task and moves to Room 3 to carry out the cleaning task assigned to the blue robot whose cleaning task was interrupted by the visitor guidance task.

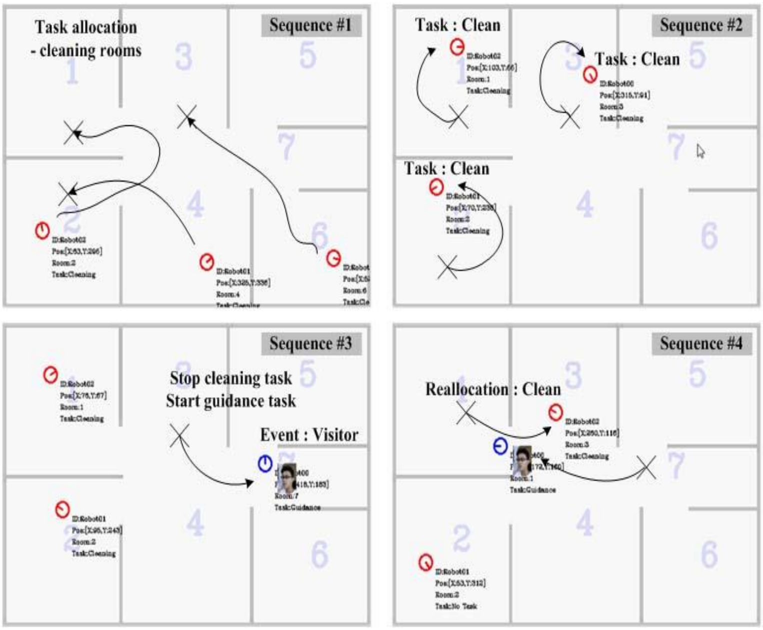



**Figure 14. Sequences in the Guidance Task Scenario**

## 5. Conclusion and Future Works

In this paper, we proposed an efficient task management algorithm for scheduling the tasks in a multi-robot system in an intelligent home space. The algorithm selects the most appropriate robot among many other robots for a certain task and reallocates robots to certain environments. The important responsibilities of the proposed algorithm are to delegate a task to a robot, reallocate a delayed task to more than one robot, and withdraw excess robots from a current task. According to the three different scenarios—guidance, cleaning, and emergency alarm—simulations are performed. In the simulation results, the proposed algorithm shows reasonable task management for home service-robots and manages the tasks of the robots dynamically.

In the near future, we intend to refine the proposed task management algorithm for better performance and extend it to more services.

## Acknowledgment

This work was supported by the Incheon National University Research Grant in 2014.

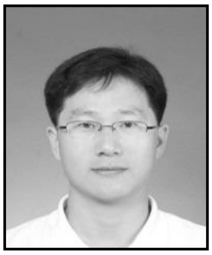
## References

[1] W. Qin, Y. Suo and Y. Shi, "CAMPS: A Middleware for Providing Context-Aware Services for Smart Space", LNCS, vol. 3947, **(2006)**, pp. 644-653.
[2] H. Kim, Y.J. Cho and S.R. Oh, "CAMUS: A Middleware Supporting Context-Aware Services for Network-Based Robots", in Proc. IEEE Workshop on Adv. Robotics and its Social Impacts, **(2005)**, pp. 237-242.
[3] J.H. Lee and H. Hashimoto, "Intelligent Space–Concept and Contents", Adv. Robotics, vol. 16, no. 3, **(2002)**, pp. 265-280.
[4] B.N. Schilit and M. Theimer, "Disseminating Active Map Information to Mobile Hosts", IEEE Networks, vol. 8, no. 5, **(1994)**, pp. 22-32.
[5] C E. Harniak, "Bayesian Networks without Tears", AAAI, **(1991)**, pp. 50-63.
[6] K. Shah and Y. Meng, "Communication-Efficient Dynamic Task Scheduling for Heterogeneous Multi-Robot Systems", in Proc. IEEE Int. Symposium Comp. Intelli. Robotics & Autom., **(2007)**, pp. 230-235.
[7] B.J. Lee, H.G. Lee, J.H. Lee and G.T. Park, "A Design of a Data Accessing Service for a Real-Time Vision Service in the Resource Sharing Architecture", in Proc. IEEE Intl. Conf. Robotics Autom., **(2002)**, pp. 1235-1240.
[8] H. Surmann and A. Morales, "Scheduling Tasks to a Team of Autonomous Mobile Service Robots in Indoor Environments", J. UCS, vol. 8, no. 8, **(2002)**, pp. 809-833.
[9] N.S. Park, K.W. Lee and H. Kim, "A Middleware for Supporting Context-Aware Services in Mobile and Ubiquitous Environment", in Proc. Intl. Conf. Mobile Business, **(2005)**, pp. 694-697.
[10] J.H. Kim, D.W. Kim, B.J. You and G.T. Park, "A Design of Framework for Smart Services of robots in Intelligent Environmnt", Int'l J. Control and Automation, vol. 2, no. 4, **(2009)**, pp. 1-12.

## Authors

**Jong W. Park**, he received the Ph.D. degree in Electronic Engineering from Inha University in 1985. He is professor of Incheon National University. He had been Director of Incheon Technology Innovation Center from 1999 to 2003. His research interests include robotics and motion estimation of robot vision system.

**Joo-Hyung Kim**, he received his B.S. degree in computer science from Korea University, Korea in 2006, and his M.S. and Ph.D. degrees in electrical engineering from Korea University, Korea in 2008 and 2013. His research areas include intelligent space, multi-robot control, machine learning, and robot vision. He joined Samsung electronics Corp. in 2013 where he is currently a senior engineer in the Mechatronics group, Manufacturing technology center.



**Dong W. Kim**, he received the Ph.D. degree in Electrical Engineering from Korea University in 2007. He is associated professor of Inha Technical College. His research interests include robotics, advanced robot design, evolutionary multi-mobile robot system, humanoid robot, soft computing and their application to control.