International Journal of Control and Automation
Vol. 2, No. 4, December, 2009

A Design of Framework for Smart Services of robots in Intelligent Environment

Joo-Hyung Kim[1], Dong-Won Kim[2], Bum-Jae You[3], and Gwi-Tae Park[1,*]

[1] *Department of Electrical Engineering, Korea University, Seoul, Korea*
*{proteus99, *gtpark}@korea.ac.kr*
[2] *Department of Digital Electronics, INHA Technical College, Incheon, Korea*
dkim@inhatc.ac.kr
[3] *Center for Cognitive Robotics Research, Korea Institute of Science and Technology, Seoul, Korea*
ybj@kist.re.kr

Abstract

This paper presents a framework for smart services of robots in an intelligent environment. Within such intelligent environment, the target platform consists of service robots and a framework for smart services including task allocation and task scheduling. For task allocation, we use an auction-based method and a knapsack problem algorithm. In this paper, the characteristics of the algorithm for allocation balancing are to delegate executing task to another robot, to reallocate the delayed task to more than one robot, and to withdraw over-allocated robots from the current task.

Keywords: *intelligent environment, smart services, task management*

1. Introduction

Recently, computer networks have developed by many researchers and have become an important part of our daily lives. Intelligent environment is one of such computer networks, and it enables to accomplish diverse services, such as visitor guidance and indoor surveillance [1]. The environment is able to monitor what is happening in it, to communicate with their inhabitants, and to make a decision something important. In [2], sensing and reacting context, information sensed to characterize the situation of the people, activities, interaction between user and application are prominent characteristic of the intelligent environment. In [3], "Intelligent Space" was proposed by Lee *et al* for interaction between human and space based on intelligent environment. In the space, a robot is used as a physical agent for offering human-centered services [4], [5]. Intelligent Space has a lot of latent abilities and advantages as a good platform to be able to offer intelligent services for both human and robots, and it can be applied to diverse areas, such as home, factory and building, in which robots can coexist with human. In such an intelligent environment, the more people require diverse services, the more complex and huge the system becomes, and the role of robots as physical agent for offering smart services becomes more and more various and important. In addition, it could be happened many events between humans and robots. And there are many sensors and robots in the same space, so that we should manage them efficiently and effectively. To perceive events occurred in such an environment and to decide tasks of robots for proper services according to the context-situation, network based service framework is needed. Thus, we need network based service framework to manage service tasks of robots.

2. Architecture for Intelligent Environments

2.1. Concept and structure

In this paper, we use a Resource Sharing Architecture (RSA) [6] as an architecture for an intelligent environment. The main focus of RSA is to share physical resources, and to organize them effectively for supporting high quality services. The structure of RSA consists of physical resources and service objects as shown in Figure 1. Physical resources are devices such as mobile robots, cameras and so on. Some of them obtain data from the environment, and transmit the data to service objects via local area network. The others provide information to human who wants to know. Whereas service objects, such as robot navigation and room cleaning service, create significant information using data transmitted from physical resources and inform the information to other services, so that an intelligent space can obtain states and abilities of physical devices connected with network. According to the relation of each service object, they are classified into three categories as follows:

1. Fundamental service: It's a service object directly connected with physical devices. Such objects obtain data from sensors, or transmit useful information to human using devices. And the services inform data to inherited service objects.
2. Inherited service: It's organized by more than one fundamental service object and other inherited service objects. It uses the results of fundamental services, and creates visual or acoustic information for transmitting to iSpace service.
3. iSpace service: It's an intelligent service object by combined diverse service objects as mentioned above (i.e. fundamental service and inherited service), which is able to carry out an intelligent task such as room cleaning or visitor guidance.

2.2. Characteristics and advantages

We explain properties of RSA briefly, and describe its characteristics and advantages. RSA should be satisfied with two properties as follows:

1. Scalability: RSA is a service-oriented architecture based on distributed sensor network. And, each service object doesn't undergo various influences by others. For the reason, although they are getting more and more increasing, the complexity of RSA doesn't almost increase, and the architecture can keep itself in good performance.
2. Flexibility: Service objects of RSA consist of a lot of software modules. From an implementation of view, software developers are easy to modify and to improve the organization of RSA. In addition, they can easily replace an existing service object with another or new one, which is not only a service object but also a physical device.

The most important characteristic of RSA is that it makes a robot overcome something physical. Although a robot doesn't have any display device, it can show people visual information using some display devices attached on wall in the space. In other words, RSA brings about a good effect which is increase of efficiency of using physical resources. In addition, one of the characteristics is that such external resources are regarded as local components on a robot itself. As the result, a robot can perceive broadly circumstances what is happening in the space.

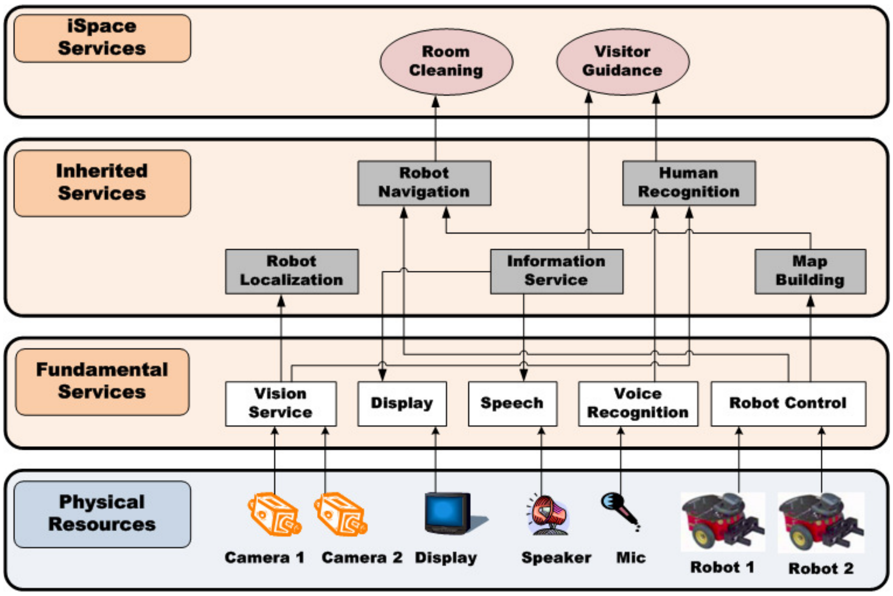


Figure 1. Resource sharing architecture

3. Framework for smart services based on RSA

In this paper, we design a multi agents based framework using a RSA. The framework consists of some robot groups and a task management group as shown in Figure 2. In a robot group, each robot agent uses some services to control each robot. In a task management group, each task management agent can know what the space should do using information obtained from the diverse services.

3.1. Robot agent

A robot agent monitors the state of a robot, and creates services to execute the assigned tasks. When a task is assigned to a robot agent, a robot starts carrying out the task using created services. If the executing task is stopped by some event, the agent informs the state of the interrupted task to a task management agent so as to be continued the task by other robot agents.

3.2. Robot group

According to the ability of each robot, they can be divided into more than one group. In case of a cooperative work, a task management agent makes a robot group, and add robots for carrying out the assigned task to the group.

3.3. Task management agent

A task management agent plans the schedule of the robot's task. To manage the tasks, the agent acquires information from some services and robot agents, and creates the task list of robots to manage assigned tasks, and allocates the tasks to 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 decided by the task management algorithm which is included in the agent. We will deal with the algorithm for task management in the following section.

3.4. 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.

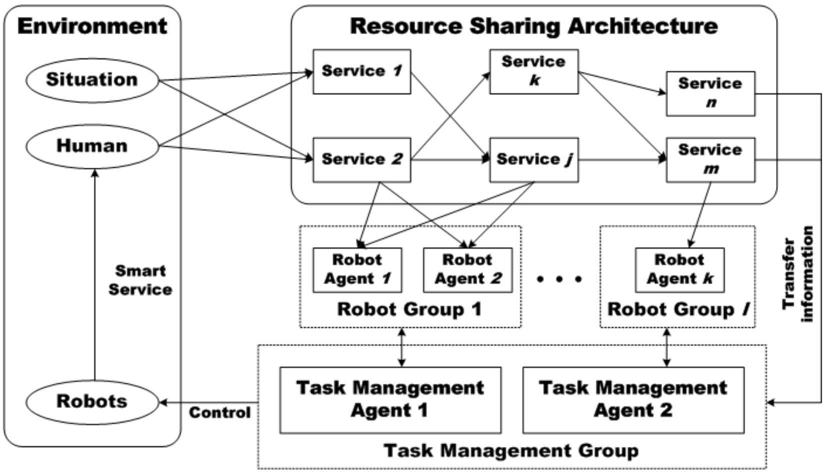


Figure 2. Overview of framework for smart services

4. Smart Service Management Algorithm for Multi-robot

In Intelligent environment, people can be provided physical service by robots and other devices. Thus, a task management agent should manage the tasks of both robots and devices. In this paper, we only focus on the task of robots, and the assumptions of tasks that we consider are described as follows:

1. Tasks can be occurred simultaneously, periodically, randomly or by human according to the state of an environment.
2. A task can be decomposed into more than one independent sub-task.
3. A task can be interrupted by an event, and the task can be continued by other robots.

Figure 3 depicts a flow of an algorithm for task management. A task management agent selects robots to carry out the assigned tasks, and informs to a robot agent. The characteristics of the proposed task management algorithm are to delegate executing task to another robot, to reallocate a delayed task to more than one robot and to withdraw over-allocated robots from the current task. In this paper, we implement two algorithms for task management of service robot as followed subsections.

4.1. How to select the best robot

To solve the problem for selecting the best robot, we consider *single-robot of tasks* and *single-task of robots*. In case of *single-robot of tasks*, the tasks are just inserted into the task list using an auction-based method [7]. However, in case of *single-task of robots*, when the decomposed tasks are inserted into the task list, we should consider effective decomposition of the task so as to distribute equally the workload of robots. In the case of a cooperation work, this approach can reduce the waiting time (i.e., idle time) to start the assigned tasks with other robots. In order to solve the problem, we apply a knapsack problem. Specially, we wish to calculate the workload $W[n,\overline{w}]$ using equation (1), so as to distribute equally the task time of all robots.

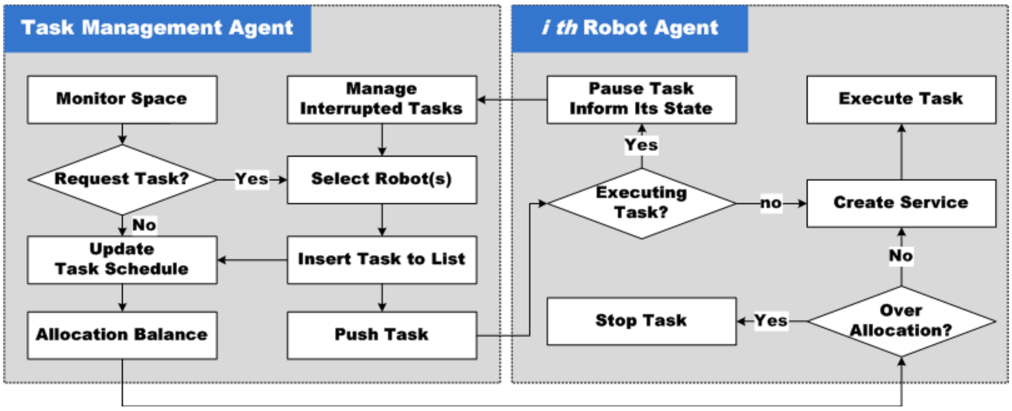


Figure 3. Flow of task management

$$\overline{w} = \frac{1}{n}\sum_{i=1}^{m} w_i,$$

$\overline{w}$: avearge task time
w_i : expected time of i th task
n : the number of robots
m : the number of total tasks included remained tasks

$$W[n,\overline{w}] = \begin{cases} \max(W[n-1,\overline{w}], w_n + W[n-1,\overline{w}-w_n]) & (w_n \le \overline{w}) \\ W[n-1,\overline{w}] & (w_n > \overline{w}) \end{cases} \quad (1)$$

where $\overline{w}$ is the average time of tasks, w_i is the expected time of ith task, and n is the number of robots. Here, we calculate all the values of the array $W[n,\overline{w}]$ using the recursive expressions above to calculate subsequent values.

4.2. Allocation balancing

In general, insufficient allocation of physical resources may bring about the task delay. On the other side, over-allocation for completing an assigned task quickly is not good. A good allocation is to complete an assigned task almost at a prearranged time and to carry out tasks as many as possible using given robots. In this paper, we consider not the number of assigned task but the deadline of assigned task for dynamic task reallocation. The objective of allocation balancing is to reallocate a delayed task to other robots and to withdraw over-allocated robot from the current task. The algorithm predicts the reallocation point x according to the progress of the assigned task. The equation for predicting is expressed as follows:

$$T = \sum_{i=1}^{x} \left[n_i \quad t_i^{'} / c \right] + \sum_{i=x+1}^{l} \left[n_i \quad t_i^{'} / (c + p) \right] \qquad (2)$$

where l is the number of total task, n_i is ith task, $t_i^{'}$ is the measured time of ith task, c is the number of allocated robots, p is the number of predicted robot for additional allocation, and x is prediction step. T includes the number of total task and the time of total task.

In case of task delay	In case of early completion

$p = 0$;
 	
do {
    p++;
    $x = \frac{T(c+a) - nc}{a}$
} *while* ($x < 1$);
 	
if ($x < n$ && $x \approx p$) {
  Add robot;
}
	$p = 0$;
	
	do {
	    p++;
	    $x = \frac{nc - T(c+a)}{a}$
	} *while* ($x < 1$);
	
	if ($x < n$ && $x \approx p$) {
	  if ($c + p < 1$) {
	      Withdraw robot;
	    }
	}

Figure 4. Task management for task reallocation

As described in Figure 4., either delayed task or early completed task, the task management algorithm calculates reallocation step and the number of predicted robot. Finally, task management agent updates the task list of robots. Next, the robot agent creates needed services to carry out an assigned task. If a robot agent gets a new task from task management agent in the middle of task executing, the robot agent stops the current task, and reports its state to task management agent. Then, the agent manages interrupted tasks, and reallocates other robot agents.

5. Case Study : Simulation for Smart Services of Multi-robot

In this paper, we evaluate the proposed algorithm through two simulation scenarios. In the simulations, there are three mobile robots. And, we set up cleaning scenario to evaluate the algorithm, and assume that the robots have same ability. In the first scenario, we compare an auction-based method with a knapsack problem to evaluate task allocation algorithm. In second scenario, we evaluate the proposed algorithm for allocation balancing according to a change of the task time.

5.1. Evaluation for selecting the most appropriate robots

The objective of this simulation is to evaluate the proposed task management algorithm using a knapsack problem whether the algorithm distributes equally the workload of robots. The scenario for evaluating is as follows:

Scenario 1
Step 1: *There are three robots. And 30 tasks will be allocated to robots. The task time is 10 or 50. Task time is generated randomly every step.*
Step 2: *By an auction-based method, a task management agent allocates 30 tasks to robots.*
Step 3: *By a knapsack problem, a task management agent allocates the same tasks to robots.*

Figure 5 represents the results of task allocation using an auction-based method and using a knapsack problem. The lines in the figure show the cumulative time of tasks assigned to robots every step. In the experimental result, our method for the balance of robot's workload shows more effective than an auction-based method.

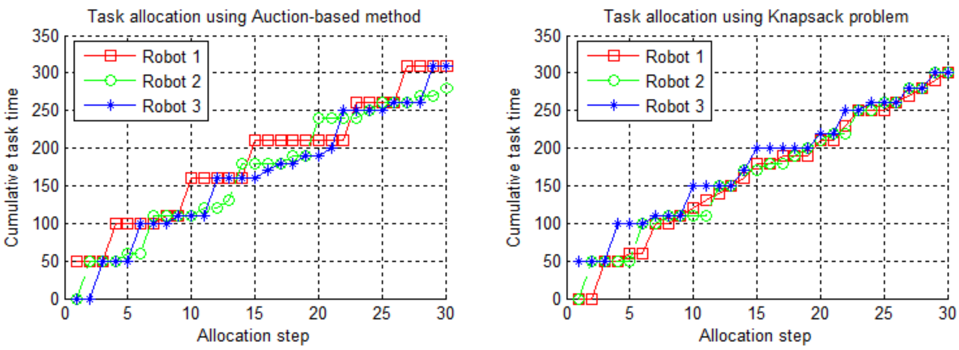


Figure 5. Comparing auction-based method with knapsack problem

5.2. Evaluation for allocation balancing

In this scenario, we evaluate the proposed algorithm which allocates robots dynamically to complete the assigned task by deadline. The scenario is as follows:

Scenario 2
Step 1: *Robot 1 and Robot 2 have to carry out Task B. However, the executing time of Task B is reduced because of the delay of Task A.*
Step 2: *According to the task schedule, a task management agent allocates Task B to Robot 1 and Robot 2.*
Step 3: *A task management agent monitors the state of Task B, and allocates the task to Robot 3 so as to complete the task within deadline.*
Step 4: *In opposition, when Task A is completed earlier than the schedule, a task management agent withdraws over-allocated robots from task B.*

At first, we measured the average time (i.e., about 43,000 seconds) of Task B, and considered two cases which are the increase of the task executing time and the decrease of the time. Table 1 and Table 2 depict the result of the measured task time and the predicted point for additional allocation according to the change of the task time. In the tables, the value of 49th / 50th means that the 49th task is allocated to Robot 3 in 50 independent tasks. And, we calculated the ratio of task delay from the result of the tables using equation (3).

Table 1. Case 1: Decrease of the executing time of the task

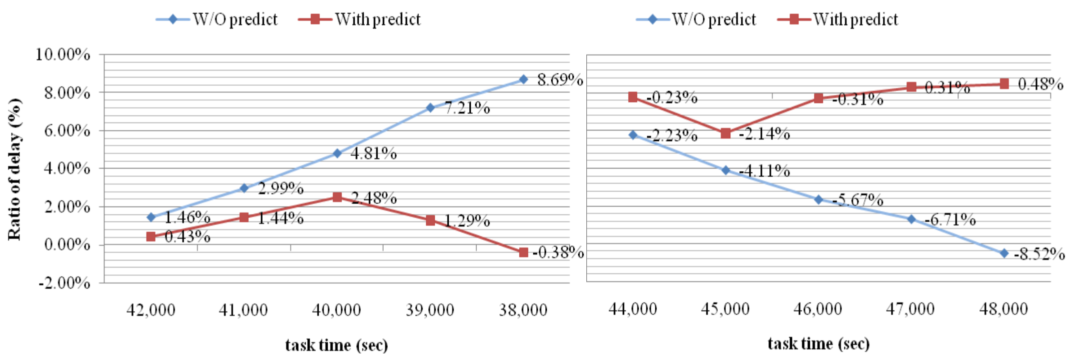
Task time	42,000	41,000	40,000	39,000	38,000
Measured time	42,182	41,588	40,994	39,504	37,857
Predicted point	49th / 50th	46th / 50th	43th / 50th	37th / 50th	36th / 50th

Table 2. Case 2: Increase of the executing time of the task

Task time	44,000	45,000	46,000	47,000	48,000
Measured time	43,897	44,038	45,859	47,143	48,229
Predicted point	47th / 50th	46th / 50th	44th / 50th	42th / 50th	41th / 50th

$$\text{Ratio of delay}\,(\%) = \frac{task\ time - measured\ time}{task\ time} \qquad (3)$$

In Figure 6, the blue-colored line represents the ratio of task delay without the prediction (i.e., the reallocation point calculated by the proposed algorithm), and the red-colored line represents the ratio of task delay with the prediction. The value of 0% means that there is no ratio of the task delay, the positive value means the task is delayed, and the negative value means the task is completed early. In the experiment result, the ratio of task delay is about 1.054% in the case 1, and it is about -0.379% in the case 2. In case that the prediction is not considered, the ratio of task delay is changed almost linearly, but the proposed algorithm shows that try to reduce the ratio of task delay in the opposite case.



(a) Decrease of the task time (b) Increase of the task time

Figure 6. Ratio of task delay

Figure 7 depicts the difference of the time which is to subtract each measured time of the task about two cases from the average time of cleaning task, and shows the variance which represents the reliability of the task completion about deadline. In case 1, the task is delayed in an average of 425.12 seconds with a standard deviation of 486.9 seconds, and the reliability is about 81.94%. In case 2, the task is completed early in an average of 166.74 seconds with a standard deviation of 458.4 seconds, and the reliability is about 87.02%. All two cases completed the task that the ratio of task delay is within about ±5%, as shown Figure 7.

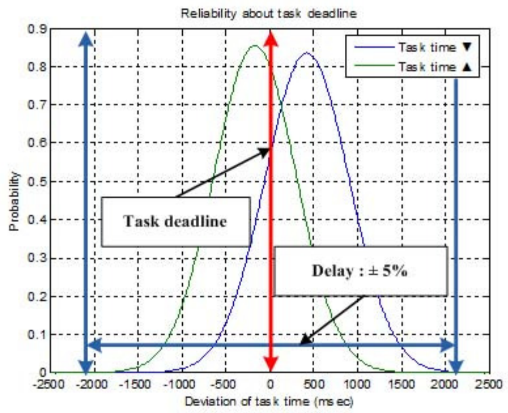


Figure 7. Reliability of task deadline

5.3. Application

We apply the proposed algorithm for task management of multi-robots to home automation. Scenario 3 shows guidance service and cleaning server.

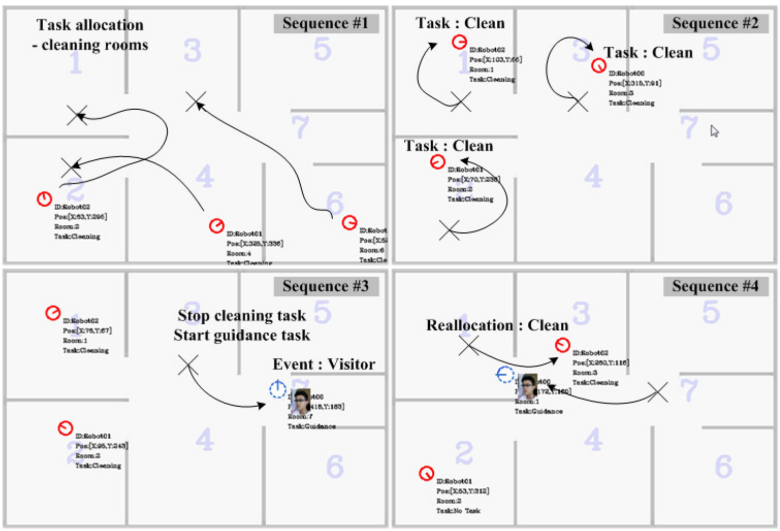


Figure 8. Scenario results for guidance service and cleaning service

Scenario 3
***Step 1**: Three robots are on standby at random position.*
***Step 2**: Cleaning task is allocated to three robots by someone.*
***Step 3**: All robots are cleaning in Room 1, 2, 3. After a few minutes, a visitor is coming.*
***Step 4**: A Space detects the visitor, and informs the event to task management agent, which decides the most appropriate robot to carry out the guidance service.*
***Step 5**: The selected robot stops the cleaning task, and move to the visitor.*
***Step 6**: The selected robot is guiding the visitor.*

Step 7*: Another robot has completed the cleaning task in Room 1, and moves to complete the cleaning task in Room 3.*

The result of this simulation is represented in Figure 8, which is the captured image in accordance with the scenario order. In Sequence #1 and Sequence #2, three red-colored circles are robots which are executing the assigned cleaning task, and a dashed line and blue-colored circle is a selected robot for visitor guidance in Sequence #3. Finally, a robot in Room 1 completes its task, and moves to Room 3 to carry out the cleaning task interrupted by a dashed line and blue-colored robot.

6. Conclusions and Future Works

In this paper, we proposed a framework for smart services of robots in an intelligent environment. The important characteristics of the proposed framework are to delegate executing task to a robot, to reallocate the delayed task to more than one robot and to withdraw over-allocated robots from the current task. In near future, we improve the proposed framework for carrying out complex services using both robots and physical devices.

Acknowledgments

This work was supported by the Korea Institute of Science and Technology (KIST).

References

[1] M. Weiser., "The computer for the twenty-first century", Sci. Amer., vol. 265, pp. 94-100, 1991.

[2] Qin, W., Suo, Y., Shi, Y., "CAMPS: A Middleware for Providing Context-Aware Services for Smart Space", *LNCS*, 3947: 644-653, 2006.

[3] Lee, J.H, Hashimoto, H., "Intelligent Space: its concept and contents", Adv. Robotics Journal, vol. 16, no. 4, pp. 265-280, 2002.

[4] Yamaguchi, T., Sato, E., Takama, Y., "Intelligent Space and Human Centered Robotics", IEEE Trans. on Industrial Electronics, vol. 50, no. 5, Oct. 2003.

[5] Lee, J.H., Morioka, K., Ando, N., Hashimoto, H., "Human Centered Ubiquitous Display in Intelligent Space", 33rd Annual Conference of Industrial Electronics Society 2007, pp. 22-27, Taipei, Taiwan, Nov 2007.

[6] Lee, B.J., Lee, H.G., Lee, J.H., Park, G.T., "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., 1235-1240, 2002.

[7] Shah, K., Meng, Y., "Communication-Efficient Dynamic Task Scheduling for Heterogeneous Multi-Robot Systems", In Proc. IEEE Int. symposium Comp. Intelli. Robotics & Autom., 230-235, 2007.

Authors

Joo-Hyung Kim

He is received the B.S. degree in computer science from Korea University, Korea in 2006, and the M.S. degree in electrical engineering from Korea University, Korea in 2008. He is now a Ph.D. student Dept. of electrical engineering, Korea University. His research areas include Intelligent Space, multi-mobile robot control, robot intelligence, and learning theory.

Dong-Won Kim

He is received Ph.D. degree in electrical engineering from Korea University, Seoul, Korea in 2007. Dr. Kim was a post-doctoral research scholar at BISC (Berkeley Initiative in Soft Computing), University of California Berkeley in 2008 and AHMCT (Advanced Highway Maintenance and Construction Technology Research Center), University of California Davis in 2009. He is now a professor in Dept. of Digital Electronics, INHA Technical College. His research focuses on the intelligent humanoid robot, autonomous multi-mobile robot navigation, and robot intelligence based on the neuro-fuzzy system. He is editorial board member of *International Journal of Control and Automation*, editor of *Journal of Convergence Information Technology, International Journal of Advancements in Computing Technology,* program committee and associated editor of several international conferences. He is also potential reviewer of IEEE Trans. Systems, Man, and Cybernetics, International Journal of Systems Science. Dr. Kim received the best paper award at international conference on knowledge-based & intelligent information & engineering systems, and Best Student Paper Competition Finalist Award at IEEE international conference on SMC in 2003, the Prize from Seoam Scholarship Foundation in 2002, and he was the prize winner of the Student Paper Contest from IEEE Korea Council in 2000.

Bum-Jae You

He is the director and the principal researcher of the Center for Cognitive Robotics Research, Korea Institute of Science and Technology (KIST), Korea. He acquired his Ph. D. from the Department of Electrical and Electronics Engineering, Korea Advanced Institute of Science and Technology (KAIST), Korea. His research areas include humanoid robots, vision-based robotics, and digital signal processor applications.

Gwi-Tae Park

He received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Korea University in 1975, 1977 and 1981, respectively. He was a technical staff member in the Korea Nuclear Power Laboratory and an electrical engineering faculty member at Kwangwoon University, in 1975 and 1978, respectively. He joined Korea University in 1981 where he is currently a Professor in school of Electrical Engineering. He was a Visiting Professor at the University of Illinois in 1984. He is a fellow of the Korean Institute of Electrical Engineers (KIEE), the Institute of control, automation, and system engineers Korea (ICASE), and advisor of Korea Robotic Society. He is also a member of Institute of Electrical and Electronics Engineers (IEEE), Korea Fuzzy Logic and Intelligent Systems Society (KFIS).