Hierarchical Planning of Modular Behaviour Networks for Office Delivery Robot

Jong-Won Yoon and Sung-Bae Cho
Department of Computer Science, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, 120-749, Seoul, Korea

Keywords: Office Delivery Robot, Hybrid Robot Control, Behaviour Networks.

Abstract: This paper proposes a hybrid architecture based on hierarchical planning of modular behaviour networks for generating autonomous behaviours of the office delivery robot. Behaviour networks suitable for goal-oriented problems are exploited for the architecture, where a monolithic behaviour network is decomposed into several smaller behaviour modules. In order to construct and adjust sequences of the modules the planning method considers the sub-goals, the priority in each task and the user feedback. It helps a robot to quickly react in dynamic situations as well as achieve global goals efficiently. The proposed architecture is verified on both the Webot simulator and Khepera II robot in office environment with delivery tasks. Experimental results confirms that a robot can achieve goals and generate module sequences successfully even in unpredictable situations, and the proposed planning method reduces the elapsed time during tasks by 17.5%.

1 INTRODUCTION

Due to the advancement of robotic technology service robots are supporting people in their daily activities (Huttenrauch et al., 2004). Especially, the mobile robots in the office environment are very helpful for users to conduct routine tasks. Several control structures for the office delivery robots have been proposed with various approaches (Beetz et al., 2001; Chung and Williams, 2003; Milford and Wyeth, 2010; Ramachandran and Gupta, 2009).

The conventional planning-based methods have been adopted to generate behaviours of mobile robots in well-known environments. They can generate the behaviour sequences optimized in predefined environments, but have the difficulty of low flexibility in complex environments. On the other hand, reactive systems can generate behaviours quickly based on environmental stimuli in complex domains (Mataric, 1998). But it also has the difficulty to generate behaviours robustly when consistency or stability is insufficient. These characteristics facilitate hybrid behaviour generation architectures of the deliberative and reactive systems.

In this line of research, we propose a hybrid architecture composed of several behaviour networks and planning method, which are regarded as the reactive and deliberative levels, respectively. For the service robot, the behaviour-based method is more appropriate because it is more important to achieve goals and maintain autonomy. In this reason, the proposed architecture exploits the behaviour networks for autonomous behaviours of the office delivery robot, which have been known as useful in goal-oriented problems (Nicolescu and Mataric, 2002; Weigel et al., 2002; Yoon and Cho, 2010; Lim et al., 2009).

In a real-world environment like office, delivery robots interact with environments and there are chances to face with various new circumstances during their tasks. To deal with these points, many researchers tried to propose the structures of office delivery robots with several different approaches.

Chung and Williams divided the original problem into several sub-problems to perform plans by reducing the complexity of the problem (Chung and Williams, 2003) and Ramachandran and Gupta proposed POMDP-based reinforcement learning for delivery robot (Ramachandran and Gupta, 2009). Some reactive methods look like similar to the proposed method that can deal with environmental changes without environmental information. But hey have the limitation to achieve only local goals and react to current exceptions without any consideration of global goals.

Yoon J. and Cho S..
Hierarchical Planning of Modular Behaviour Networks for Office Delivery Robot.
DOI: 10.5220/0003982100140020
In *Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2012), pages 14-20
ISBN: 978-989-8565-22-8
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

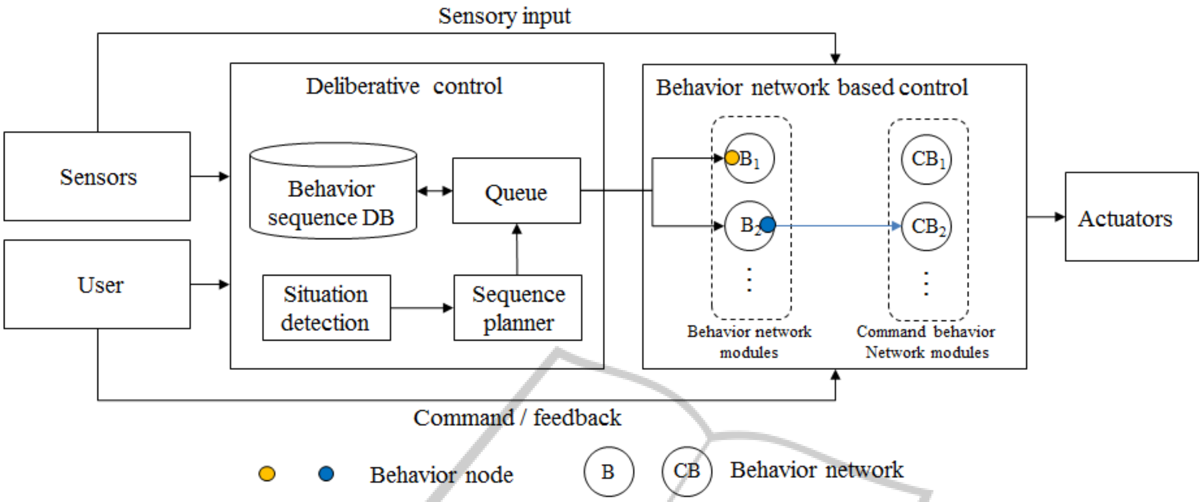

Figure 1: Architecture of the proposed hybrid behaviour network system.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

To work out this problem, some hybrid architectures have been proposed. Milford and Wyeth used different obstacles and experience maps for local and global navigations, respectively (Milford and Wyeth, 2010). The method used low-level controls for reactive actions that were managed by high-level controls. The proposed method is based on reactive approaches because it mainly use behaviour networks but the planning is externally placed at higher level to control them dynamically by considering the global goals in order to overcome the limitations of conventional reactive methods.

2 HYBRID ARCHITECTURE

The proposed architecture for the autonomous office delivery robot to generate behaviours consists of two levels. Lower level includes behaviour network-based modules which can reflect temporary environmental changes, and upper level, a deliberative system, controls the goals and plans flexibly according to situations.

Figure 1 shows the proposed architecture of the hybrid behaviour network system. The behaviour network-based control includes the specific behaviour networks and the common behaviour networks, and the deliberative plan control.

2.1 Behaviour Network Modules

Contrary to the conventional reactive systems, the behaviour network not only generates behaviors instantly but also has goals, with which can solve some simple planning problems. However, as the problem gets more complex, it is difficult to select behaviours accurately with only one monolithic network (Decuqis and Ferber, 1998; Tyrell et al., 1993). In order to overcome this shortcoming, the behaviour network is divided into several modules.

The objectives of the modularized behaviour networks are as follows.

- The modular behaviour network is easier to be designed and reused than one monolithic network (Nicolescu and Mataric, 2002).
- Confusions which can be occurred when selecting behaviors in one large flat network can be reduced by giving only one goal to each smaller network module (Tyrell et al., 1993).

Each module in the proposed architecture has a behaviour network oriented to single corresponding goal. The behaviour network is used as the method for selecting the most natural and suitable behaviours for the situations. The behaviour networks are the model that consists of relationships between behaviours, a goal, and external environment, and selects the most suitable behaviour for the current situation.

In the behaviour network, behaviours, external environments and internal goals are connected with each other through links. Each behaviour contains preconditions, an add list, a delete list and an activation. The preconditions are a set of conditions that must be true in order to execute behaviours. The add list is a set of conditions that are highly likely to be true when behaviours are executed. The delete list is a set of conditions that are likely to be false when the behavioural entities are executed. The activation represents to what extent the behavioural entity is activated.

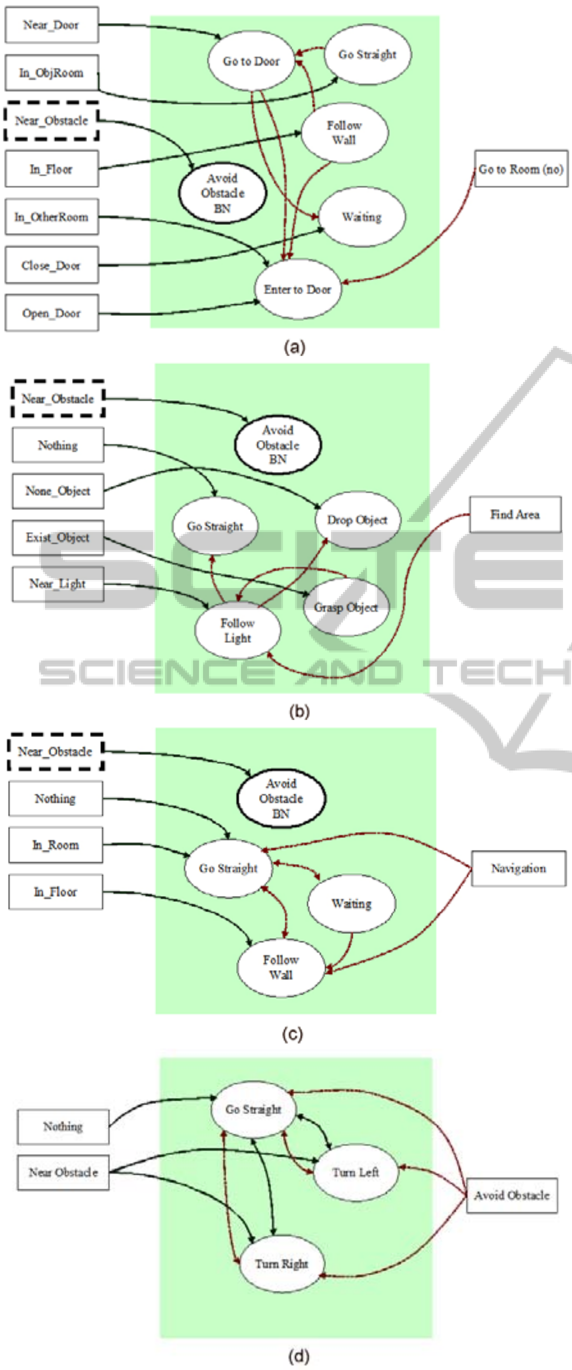

Figure 2: The behaviour networks designed.

The activation energies of behaviours firstly induced from external environments and the goal. The activation of the ith behaviour A_i can be presented as follows:

$$A_i = A_i + \sum_n w_e E_{i,n} + \sum_m w_g G_{i,m} \quad (E_{i,n}, G_{i,m} = 0, 1) \qquad (1)$$

where w_e and w_g are the weights to induce activation energies from environments and goal respectively. $E_{i,n}$ and $G_{i,m}$ represent whether the nth environment element and the mth goal are connected with the ith behaviour or not, respectively.

After the first induction, behaviours exchange their activation energies with other behaviours considering the type of their links. The behaviour exchange can be presented as follows:

$$A_i = A_i + \sum_n (w_p P_{i,j} + w_s S_{i,j} - w_c C_{i,j}) \quad (i \neq j, P_{i,j}, S_{i,j}, C_{i,j} = 0, 1) \qquad (2)$$

where w_p, w_s and w_c are the weights to exchange activation energies through predecessor, successor and conflictor links, respectively, and $P_{i,j}$, $S_{i,j}$ and $C_{i,j}$ represent whether the ith and jth behaviors are connected by each type of links, respectively.

The behaviour networks have a threshold to decide which behaviours are executable. Using this, the behaviour networks select the behaviour where all the preconditions are true and the activation energy is larger than the threshold. Unless any behaviour is selected, the behaviour selection system constantly reduces the threshold until a behaviour is selected.

A behaviour network module consists of one goal, external environments, and behaviour nodes. Each module is mapped to a sub-goal from the planning system. If the planning system chooses a single sub-goal to achieve, the corresponding behaviour network module is activated and generates behaviour sequences.

In this paper, we designed two behaviour network modules–go to a room and find objects–and two common modules–navigate and avoid obstacles. Figure 2 shows the behaviour network modules designed.

2.2 Planning of Goal Sequences

In the deliberative control, the system does not plan sequences of all primitive behaviours or trajectories, but plans the sequences of sub-goals to control behaviour network modules. Since we designed several small independent behavior modules with sub-goals, they should be controlled explicitly to achieve the global goal. To plan goal sequences, the deliberative module and the behaviour network-based modules are connected.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Since the behaviour networks do not have any information about the map of the environment, it is difficult to perform plans correctly in complex environments. To deal with this, the deliberative module checks accomplishments of sub-goals and controls plans when situations are changed, and the plan in each behaviour network module controls only partial behavior sequences to achieve the sub-goal of the corresponding module.

The deliberative control module makes plan by deciding priorities of goal sequences to achieve the global goal and adjusting priorities when exceptions or feedbacks are occurred. The module uses the basic behavior library that includes basic sequences of behaviors required to perform when tasks are given. The library is defined before the usage, and can be modified by the feedbacks of the user. When the user gives tasks, the sequences are planned by using the library and inserted into the queue. At the ‘Check event’ stage, the robot checks changing of situations, and adjusts the sequences.

2.2.1 Priority-based Sequence Planning

To plan and adjust the sequences, the priorities of tasks are used. In this paper, the priority is defined as the deadline of the delivery required by the user. For this process, we define several parameters as follows:

- $C = \{c_i\}$: command set
- $D = \{\{d_i\} : \{d_i\} \in C\}$: decomposed command set
- $Q = \{q_i : q_i = d_1, ..., d_k, i < \max_{\text{queue}}\}$: command queue
- $X = \{Wait, Critical, Minor\}$: user feedback set

Firstly, priorities are determined according to the requested deadline and the order of tasks as shown below:

$$P^i_{Fix} = \frac{(t_{Max} - t_i)}{t_{Max}} \times 10 + (O_{Max} - O_i) \tag{3}$$

where t_i and O_i indicate the remaining time and the order of the *i*th task, respectively. *Max* means the possible maximum value of the corresponding variable.

Secondly, priorities are adjusted by additionally considering the position of the robot as follows:

$$P_{Dynamic}(q_i) = \begin{cases} P_{Fix}(q_i), & \text{if } From(i) = S \text{ or } t_i < \theta \\ P_{Fix}(q_j), & \text{if } From(i) \neq S \text{ and } \exists j \cdot \ni \cdot From(j) = S \\ f(S), & \text{if } S \in X \end{cases} \tag{4}$$

where S is the current state of the robot, $From(i)$ indicates the starting point of the *i*th task, and $f(S)$ is the priority decided by the feedback.

2.2.2 Sequence Queue and User Feedback

The sequence queue contains feedbacks from the user. Each of them consists of an index of the user, a type of command, a deadline, a point of departure, and a destination. When the feedback is given, the robot seeks sequences for the corresponding command and puts the sequences into the queue. If there is no relevant sequence in the library, the robot requests feedbacks to the user.

The priorities of behavior modules in the sequence are computed with the order of the task and the deadline by using Eq (3) and (4) in the section 2.2.1. Each module is sorted by the priority in the sequence queue. For this job, the queue has information. The front four are input by the user, and next five are used to manage the plan flexibly.

Each task has the segmented sequence with subtasks. For example, a single delivery task is split into the subtask to bring the object from the point of departure and another subtask to move the object to the destination. Each task has a check point that indicates which subtask is performed lastly. The check point enables to adjust the plan flexibly according to the change of situations. The subtask has the sequence of several behaviour modules.

Task adjustments are preceded according to the position of the robot as follows :

$$\begin{gathered} Seq(q_i) = \begin{cases} q_k, & \text{if (CASE1)} \\ q_j \to d_l, & \text{if (CASE2)} \\ q_i, & \text{otherwise} \end{cases} \\ (CASE1) \\ \exists \mathrm{Pos}_k \cdot \ni \cdot Pos_k = S \text{ and } t_i > \theta \\ (CASE2) \\ \exists \mathrm{Pos}_j \cdot \ni \cdot Pos_j = S \text{ and} \\ (q_j \to d_l = Take \text{ or } q_j \to d_l = Give \text{ and } \exists obj) \end{gathered} \tag{5}$$

where $Seq(qi)$ indicates the target command to be placed instead of q_i, Pos_i is a set of positions that q_i contains, and $q_i \to d_l$ is the *l*th behavior in q_i. For example, the robot may pass the other room not required for the task during the movement from the starting point to the destination. In this case, it searches the task which the robot should fulfill at its current location. If the deadline of the task in progress is greater than the threshold, it changes the plan to execute the task found with high priority. Otherwise, it ignores the task found and continues its previous job.

Table 1: Given seven delivery tasks.

Task	1	2	3	4	5	6	7
Deadline	1	1	2	3	1	2	1
Departure (Room #)	1	3	2	3	1	3	4
Destination (Room #)	2	1	3	1	4	2	1

3 EXPERIMENTS

In order to show the usefulness of the proposed architecture, we performed experiments for the office delivery tasks of the mobile robot.

3.1 Experimental Setup

The hybrid behaviour generation system is applied to the mobile robot, Khepera II, which has a wireless camera sensor, eight infra-red sensors, eight light sensors, one gripper and two motors. The experiments were performed on both the Webot simulation environment and a real-world environment.

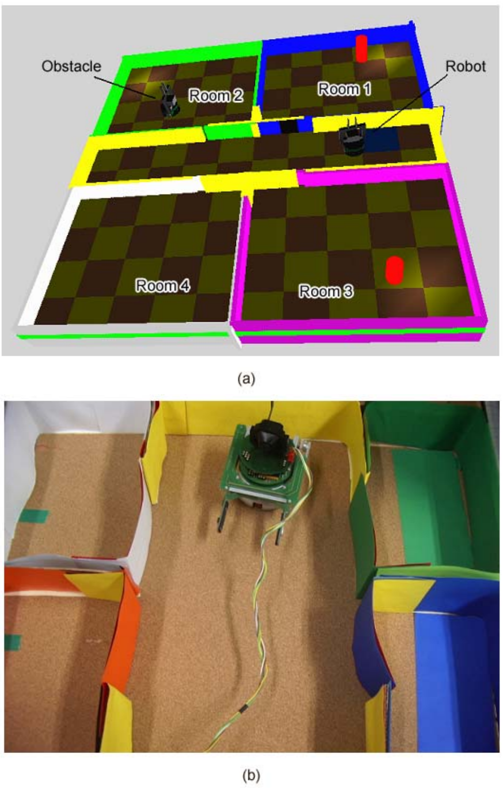

Figure 3: The experimental environment with four rooms and a corridor. (a) simulation, (b) real robot.

For the office delivery tasks, we designed the office environment which includes four rooms and one aisle. The colors of each pair of the door and the room were colored identically; therefore, the robot can recognize each room by referring the color of the corresponding door. If some doors had been closed, we changed colors of them as blacks. Since the robot does not have any information about the environment, it should navigate with only recognized colors of rooms. Figure 3(a) and (b) show the experimental environment that we constructed in the simulator and real-world, respectively.

3.2 Qualitative Analysis

In this section, we analyzed planned goal sequences from various tasks. We obtained the rates of success and failure after performing all tasks, and analyzed changing of the sequences according to errors and feedbacks from the user.

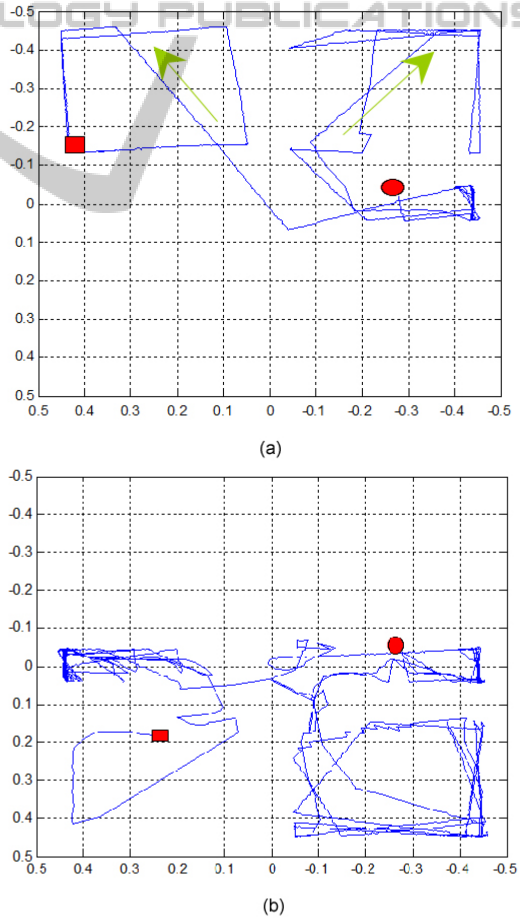

Figure 4: Trajectories of the robot.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 2: Minimum, average, and maximum steps after 30 tasks.

Minimum	Average	Maximum
804	1,930	5,370

The task of delivering the object from the specific room A to another room B was given for the experiments. First of all, we obtained the trajectories of the robot during the task. Figure 4(a) and (b) are the trajectories for the delivery task from the room 2 to the room 1 and the task from the room 4 to the room 3, respectively.

If the robot had been located in the room or at the corridor, it started the behavior module for searching the destination and used camera for sensing since it did not have map information of the environment. When the robot reached the destination room, it followed the light to find the object.

Additionally, in order to verify the usefulness of the sequence adjusting process, we designed seven delivery tasks shown in Table 1. Experiments were conducted both with and without sequence adjustments using the tasks. Sequences of chosen modules and robot's location were obtained.

With sequence adjustment processes, the robot modified its behavior sequence according to its location. If the robot achieved its goal in the certain room, it sought the task which can be started at the room. As the result, it reduced steps wasted at the corridor. The robot finished all the tasks within 3,956 steps without sequence adjustments, but it completed within 3,264 steps, 17.4% reduced, with adjustment processes.

3.3 Quantitative Analysis

For quantitative analysis, we obtained the elapsed time during tasks. We initially located the robot randomly and made it to repeat random delivery tasks 30 times. Table 2 shows minimum, average, and maximum steps after tasks.

Figure 5(a) and (b) show the trajectories obtained from results with maximum and minimum steps, respectively. The task from the room 4 to the room 2 took the smallest steps. Otherwise, the maximum steps were taken in the case that the robot was initially located at the corridor because it took long time to find the target room according to the state of the sensors. Even though the robot started the task at the corridor, differences between results were shown in accordance with the distance between the room and sensory states.

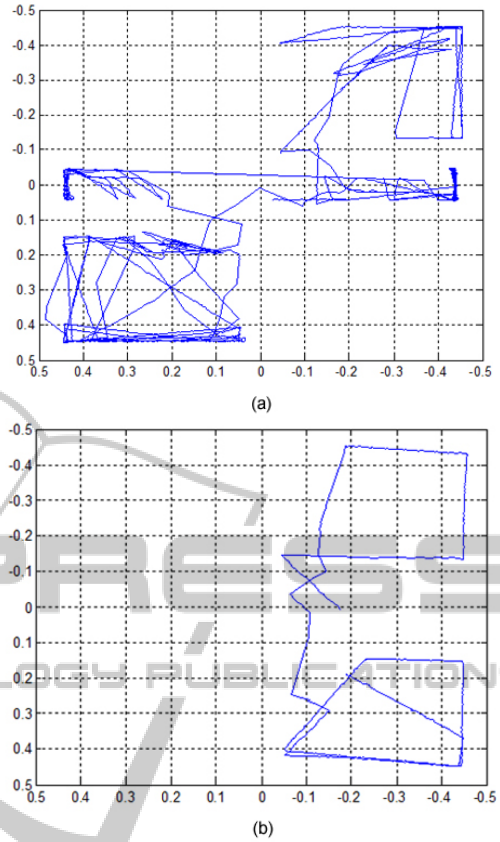

Figure 5: Trajectories from results with (a) maximum and (b) minimum steps.

4 CONCLUDING REMARKS

We proposed a hybrid behaviour system for an autonomous mobile robot for office delivery tasks. The system is oriented to the behaviour network modules which is useful to perform tasks in real-world environments. Moreover, a method for planning is attached to supplement them. The planning system generates and manages overall sequences of behaviour modules, and the behaviour modules achieve several sub-goals by generating autonomous behaviours quickly.

Experiments were conducted to verify the usefulness of the proposed architecture. We implemented a simple office environment in both the simulator and the real-world with the Khepera II mobile robot, and designed several delivery tasks. As the result, it is confirmed that the robot can achieve the goal even though there are temporary exceptions, and it changes its plan when adjustments are required to complete tasks more efficiently.

For the future works, the method for learning structures of networks and controlling them automatically should be investigated. Moreover, the proposed architecture should be tested on more realistic problems.

ACKNOWLEDGEMENTS

This research was supported by the Original Technology Research Program for Brain Science through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0018948). The authors thank for the valuable assistance to Ms. H.-J. Min who was a member of Soft Computing Laboratory, Yonsei University.

REFERENCES

H. Huttenrauch, A. Green, M. Norman, L. Oestreicher, and K. S. Eklundh, "Involving users in the design of a mobile office robot," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 2, pp. 113-124, 2004.

M. Beetz, T. Arbuckle, T. Belker, A. B. Cremers, D. Schulz, M. Bennewitz, W. Burgard, D. Hahnel, D. Fox, and H. Grosskreutz, "Integrated, plan-based control of autonomous robots in human environments," *IEEE Intelligent Systems*, vol.15, no.5, pp. 56-65, 2001.

S. H. Chung and B. C. Williams, *A Decomposed Symbolic Approach to Reactive Planning*, Master's Thesis, MIT, 2003.

M. Milford and G. Wyeth, "Hybrid robot control and SLAM for persistent navigation and mapping," *Robotics and Autonomous Systems,* vol. 58, no. 9, pp. 1096-1104, 2010.

D. Ramachandran and R. Gupta, "Smoothed SarSa: reinforcement learning for robot delivery tasks," In *Proc. of Int'l Conf. on Robotics and Automation*, pp. 2125-2132, 2009.

M. J. Mataric, "Using communication to reduce locality in distributed multi-agent learning," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 10, no. 3, pp. 357-369, 1998.

M. N. Nicolescu and M. J. Mataric, “A hierarchical architecture for behavior-based robots,” In *Proc. of First Int’l Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pp. 227-233, 2002.

J.-W. Yoon and S.-B. Cho, "A mobile intelligent synthetic character with natural behavior generation," In *Proc. of Int’l Conf. on Agents and Artificial Intelligence*, pp. 315-318, 2010.

S.-S. Lim, J.-W. Yoon, K.-H. Oh, and S.-B. Cho, "Gesture based dialogue management using behavior network for flexibility of human robot interaction," *IEEE Int'l Symp. on Computational Intelligence in Robotics and Automation*, pp. 592-597, 2009.

V. Decuqis and J. Ferber, "An extension of Maes' action selection mechanism for animats," In *Prof. of Fifth Int'l Conf. on Simulation of Adaptive Behavior*, vol. 5, pp. 153-158, 1998.

T. Tyrell, *Computational Mechanisms for Action Selection*, PhD Thesis, University of Edinburgh, 1993.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS