

Design and Development of a Social Robot Framework for Providing an Intelligent Service

Joohee Suh and Chong-woo Woo

Abstract— Intelligent service robot monitors its surroundings, and provides a service to meet a user's goal. This service could be enhanced or optimized, if it collaborates with other robots in a group. But, it is difficult to make such specific group from the many heterogeneous robots that are different in their functions and structures. This paper describes a social robot framework that outlines a multi-layer structure, which is suitable to make a particular group of robots to solve a given task by collaborating with other robots. Simulated experimentation for grouping is done by generating several heterogeneous robot instances. And the collaboration is done by multi-level task planning mechanism. The result is significant, and further experimentation with physical robot is needed.

Index Terms— Sociable Robot, Robot Grouping, Collaboration, Task Planning, Context-Awareness.

I. INTRODUCTION

The context in context awareness computing is any entities that affect interaction between human and computing devices in the environment. An entity can be a human, place, or any object relevant to the interaction between human and application [1]. Intelligent robot is an autonomous and dynamic object in this environment, and it becomes explosively interesting research area. It interacts with various computing devices, recognizes context, and provides appropriate service to human and the environment [2]. Essentially, the robot is required the two main issues, understanding and carrying out the context. Understanding context can be done in many different ways, but the general procedure includes, perceiving raw information through sensors, modeling and reasoning steps follows right after preprocessing of the raw data [3]. The procedure needs to be further studied in detail to make the system more efficient, and there are large amount of studies are being reported [1][2][3]. The second issue is that the robot has to carry out the task to meet the user's goal. But the service in this environment is often too complex that one single robot may not handle all the physical and control elements. In other words, a group of robots are needed to accomplish given task by collaborating each other. We can define this activity as a robot grouping, and we need to study further to make better group by considering their characteristics of the robot and the task.

Manuscript received July 7, 2009

This research was supported by the Seoul R&BD program of Korea.

Ms. Joohee Suh is in M.S. program with the School of Computer Science, Kookmin University, seoul, korea (first author, email: crazyDMP@gmail.com)

Dr. Chong-woo Woo is with School of Computer Science, Kookmin University, seoul, Korea (corresponding author, email: cwwoo@kookmin.ac.kr)

In this paper, we are describing a design and development of the social robot framework for providing an intelligent service with the collaboration of the heterogeneous robots, in the context awareness environment. The framework is designed as multi-layers suitable for understanding context, grouping robots, and collaborating among the robots. In this study, we mainly focused and implemented on the grouping and collaborating parts of the system. The context awareness part of the system is designed and will be implemented for the next study with ontological knowledge representations with modeling and reasoning mechanisms.

II. RELATED WORKS

A. Context-Awareness Systems

With the increase of mobile computing devices, the pervasive systems are getting popular in recent years. One part of the pervasive system is the context awareness system, which is being studied explosively in various directions. Among the many research results on this issue, the most prominent systems are the CoBrA (Context Broker Architecture) [4], SOCAM (Service Oriented Context-Aware Middleware) [5], Context-Toolkit [6]. The CoBrA and SOCAM used ontological model that supports easy sharing, reusing, and reasoning of context information. Also it supports the extensibility and interoperability. Therefore, the ontology becomes getting popular solution to represent the data for the recent context awareness systems. Short reviews of the previous systems as follows.

Context-Toolkit—this early context awareness middleware system gains information from the connected devices. But, since it does not use ontology, the standardized representation for the context is impossible, and the interoperability between heterogeneous systems is also impossible.

CoBrA—because this system is based on the ontology model, the standardized representation for the context is possible. But, since the use of ontology is limited only to a special domain, so called 'Intelligent Meeting Room', it does not guarantee any extensibility to the other diverse domains.

SOCAM—this system is based on Service-oriented structure, which is efficient middleware system for finding, acquiring, analyzing context information. But, since it depends on OWL (Web Ontology Language) for reasoning, its reasoning capability is limited to its own learning module and inference engine.

In our study, we need to overcome the limitations and will

adapt the merits of these previous works, such as, the limited standardized representation or extensibility. Since the purpose of developing our system is to accomplish proper robot actions based on the context information, we have designed the context-awareness layer adapting the COCON model [7] to overcome the limitations of the previous systems.

B. Robot Grouping and Collaboration

First of all, our research issue focused on the collaboration among the robots, which requires a group. Therefore, we first need to develop a method of grouping robots for a given task. Study on the robot grouping is just beginning, but some related researches are being reported as follows.

For instance, Rodic and Engelbrecht [8] studied initial investigation into feasibility of using ‘social network’ as a coordination tool for multi-robot teams. Under the assumption of multi-robot teams can accomplish certain task faster than a single robot, they proposed multi-robot coordination techniques. Inspired by the concept from the animal colony, Labella et al., [9] showed simple adaptation of an individual can lead task allocation. They developed several small and independent modules, called ‘s-bots’, and the collaboration in this system is achieved by means of communication among them. They claimed that individuals that are mechanically better for retrieval are more likely to be selected. Another point of view for the collaboration is task allocation among the multi-robot colony. Mataric et al, [10] had an experimentation comparing between a simulated data with physical mobile robot experiment. The result showed that there is no single strategy that produces best performance in all cases. And other approaches are the multi-robot task allocation by planning algorithm [11][12][13].

All of these research efforts are being done in many different fields, and the selection of individual is rather random or simple, which may often result in inadequacy of performing a given task. Using ‘entropy’ of information theory [14] could be a good alternative compare to the other informal approaches. Goodrich [15] argues that the behavioral entropy can predict human workload or measure of human performance in human robot interaction (HRI) domain. Balch [16] demonstrated successfully in his experimental evaluation of multi-robot soccer and multi-robot foraging teams. In our study, we will use the ‘entropy’ metric for selecting an appropriate robot from the robot colonies by generating decision tree first, to minimize the complexities of adapting the entropy.

III. DESIGN OF THE SYSTEM

As in the Fig.1, the overall architecture of our system is divided into three main layers, context-awareness layer, grouping layer, and collaboration layer. Also, we have two more sub-layers, Physical and Network layer, which will not be discussed here since they are not the main issue here. The overall process of the three main layers works as follows, and entire structure of the system can be viewed in Fig. 2.

- ① Context-awareness layer generates a context from the raw information, and then does modeling and reasoning in order to aware of the context.
- ② Grouping layer creates decision classifier based on the entropy metric, and makes a necessary group.

- ③ Collaboration layer does multi-level task planning; high-level task planning generates a set of tasks for the context, and the low-level task planning generates a set of actions for the task.

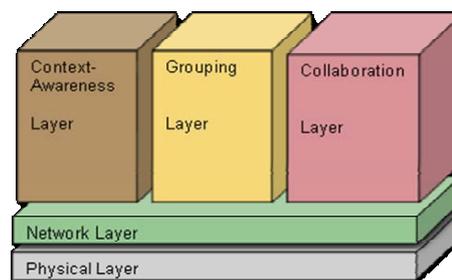


Fig. 1. General Structure of Social Robot

A. Context-Awareness Layer

The context awareness layer receives raw data from the computing devices including RFID, Zigbee, and so on, and generates a situation. It consists of four sub modules as follows.

Raw Data Collector—it receives raw data from the Physical layer passes it to the *context provider*.

Context Provider—it receives raw data from the *raw data collector* and transforms the data into *standardized context* according to low context model.

Context Integrator—it receives standardized context from the *context provider* and generates inference level context through the high-level context modeling.

Situation Recognizer—the process through the above three sub-components in sequence generates *situation(s)* by using the rule-based inference engine. This situation is delivered to the *grouping layer*, to make a group.

B. Grouping Layer

The server of the system receives information about which robot is connected to the server, and stores this information into *Group info* database. The grouping layer consists of three sub modules, *Situation Acceptor*, *Classifier Generator*, and *Groupier*.

Situation Acceptor—it receives a *situation*, from the Context awareness layer, and requests to the *Classifier Generator* to make a group of robots for this situation.

Classifier Generator—its main function is to generate a Classifier (i.e. decision tree) to make a group for a specific situation. In this study, we generate the Classifier based on ID3 decision tree algorithm.

Groupier—the robot grouping layer has two sub-modules; the *searcher* requests current situation to the connected robots, and the *classifier* generates decision tree for the acquired situation. The generated group information is stored in the *group info* repository, and will be used for collaboration.

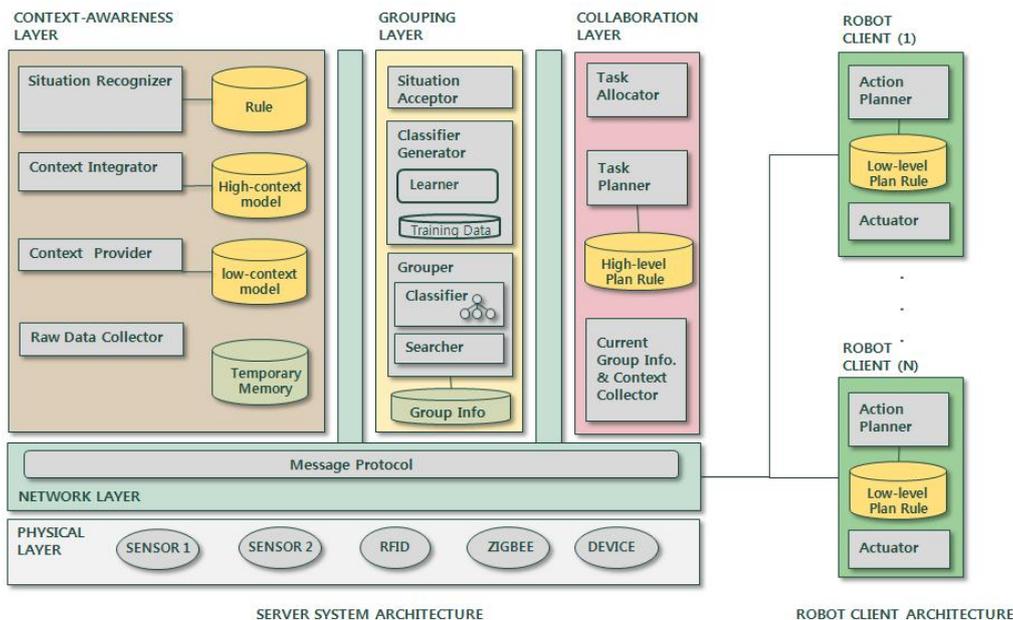


Fig. 2. Structure of Sociable Robot of the Context awareness Environment

1) Context information for the Robot

In this experiment, we designed virtual situations, such as ‘cleaning’ situation, ‘delivery’ situation, ‘conversation’ situation, and so on. And we also designed the robot has 5 different attributes, such as, ‘power’, ‘location’, ‘speed’, ‘possession’, and ‘IL (intelligence level)’. These choices can be selected arbitrarily through the user interface as in the Fig. 3 below.

From the Fig.3, we can see that the robot instances and the situation can be selected in the 3 sub-windows.

- ① ‘Set Instance’ window: can enter 5 attributes for a robot instance.
- ② ‘Control’ window: start/stop button is a request for connection to server
- ③ ‘Set & send’ window: set button can choose a virtual situation, and send button sends the situation to server.

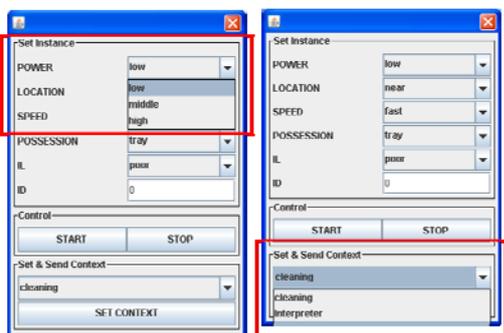


Fig. 3. User Interface for entering choices for Robot attributes and Situation

2) Training data

We are not going to use the physical sensor data, but we are going to use the arbitrarily generated simulated data. And this is done by selecting through the user interface in the Fig. 3.

Before we begin, we need to train the system first. Therefore, we prepared a training set with 10 robot instances

for creating initial decision tree. Table 1 is a set of training data that shows 10 robot instances for ‘cleaning’ situation.

Table 1. Training data to create a Decision Tree

| Idx | Power | Location | Speed | Possession | IL | Cleaning |
|-----|--------|----------|--------|-------------|--------|----------|
| 1 | Low | Near | Fast | Tray | Poor | Yes |
| 2 | Low | Far | Slow | Gripper | Normal | Yes |
| 3 | Middle | Near | Fast | Interpreter | Smart | No |
| 4 | High | Near | Fast | Tray | Normal | No |
| 5 | Low | Far | Normal | Tray | Smart | Yes |
| 6 | Low | Near | Normal | Interpreter | Normal | Yes |
| 7 | Middle | Far | Normal | Interpreter | Normal | No |
| 8 | High | Near | Slow | Gripper | Smart | Yes |
| 9 | Middle | Near | Fast | Interpreter | Smart | No |
| 10 | High | Far | Slow | Gripper | Poor | Yes |

3) Decision Tree Generation

The following equation is the entropy algorithm of the information theory, which will be used to generate a tree in our approach.

$$Entropy(S) = -P_{(+)} \log_2 P_{(+)} - P_{(-)} \log_2 P_{(-)} \quad (1)$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{S} Entropy(S_v) \quad (2)$$

We can compute the general entropy using the equation (1), and compute the gain entropy to select a single attribute using equation (2)

C. Collaboration Layer

It consists of following 3 sub-components on the server side, and action planner is on the client (robot) side. The distinctive features are as follows.

Group Info.&Context Collector: It collects information of ‘situation’ and information for ‘grouping robots’ from the *Grouping Layer*.

Task Planner (server side)—It generates a set of tasks using high-level planning rules (a global plan). For instance, the generated tasks for “cleaning” situation can be the “sweeping” and “mopping”.

Task Allocator— the task allocator sends the generated tasks to appropriate robots.

Action Planner (client side) —Generated tasks by the *task planner* is delivered to the client (robot), and further refined into set of actions by the action planner.

1) *Multi-level Task Planning*

In collaboration layer, the task is carried out by the multi-level task planning mechanism as follows (see Fig. 4).

- ① The task planner gets the situation(s), and generates a set of tasks based on the high-level planning rules.
- ② Then the system allocates the tasks to the appropriate robot who can handle the specific job
- ③ When the task allocation is done, then each assigned robot activates the action planner to generate a set of actions

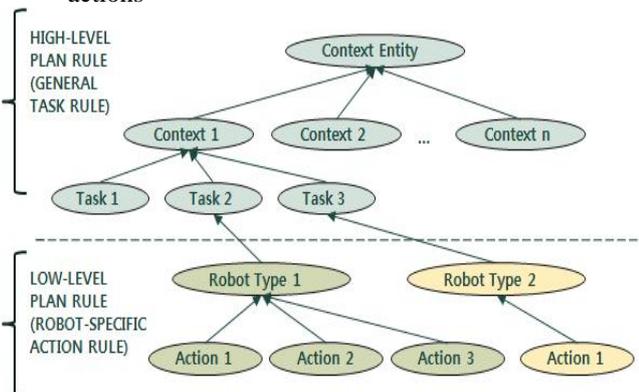


Fig. 4. Multi-layer Planning Mechanism

IV. SIMULATED EXPERIMENTATION

The experimentation is divided into two parts, robot grouping and robot collaboration.

A. *Robot Grouping*

In this study, the robot grouping experiment begins with generating virtual robot instances and selecting a virtual situation. And we generated a virtual situation with 8 robot instances as in the Fig. 5.

The Fig. 6 shows all the necessary information for grouping.

- ① ‘ Context’ window : shows selected virtual context,
- ② ‘ TrainingData’ window: shows 10 training data for selected situation.
- ③ ‘ RobotInstance’ window: shows activated instances.
- ④ ‘ Robot Grouping Result’ window: shows grouped robots as instance numbers.
- ⑤ ‘ Tree’ windows: shows the generated decision tree for the context.
- ⑥ ‘ Process’ window: shows the entropy computation process for the tree.

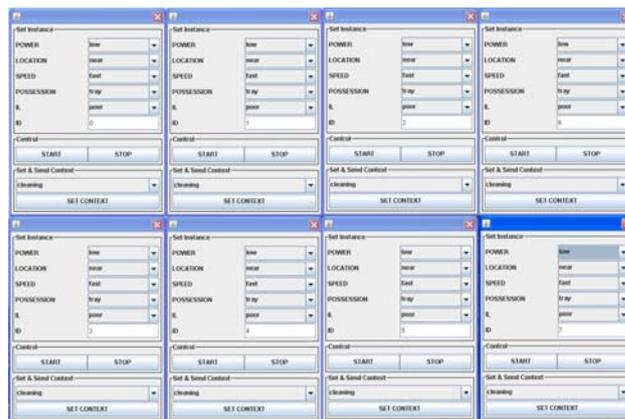


Fig. 5. Created 8 Heterogeneous Robot Instances

The entire grouping process can be summarized as follows.

- ① Create arbitrary number of robot instances through the user interface (see Fig. 3), and this information is sent to server.
- ② Also, information of a ‘situation’ is sent to server
- ③ Then, the server creates decision tree using the information of the robot instances, and decide a group of robots for the specific situation (ex. Cleaning).
- ④ As a result of the final process, the grouped robot instances are shown in the bottom left window as robot instance numbers.

From the bottom left of the windows in the Fig. 6, we can see that the robot instance number 1,4,6 and 7 is grouped together to perform given task, ‘cleaning’.

B. *Robot Collaboration*

1) *Task Planning Rules*

The task planning rules for collaboration layer is divided into 2 levels, high level planning rules (general task rules), and the low-level planning rules (robot-specific action rules). The general task rule can generate set of tasks, and the robot specific action rule can generate a set of actions to perform the task. For example, if ‘cleaning’ is the selected context, then task planner generates a set of tasks for the ‘cleaning’ as ‘sweeping’, and ‘mopping’. When the task ‘sweeping’ is assigned to a robot, the generated action plan is ‘move’, ‘lift’, ‘sweep’, and ‘release’. The task planner works on the server side, and the action planner is located on the client side. The sample task planning rules are as in the table 2.

Table 2. Task Planning Rules

| High-level Planning Rule | Low-level Planning Rule |
|---|---|
| IF (Context = “Cleaning”) THEN (needTask (Sweeping, Mopping)) | IF (Task = “Sweeping”) THEN(needAction (Move, Lift, Sweep, Release)) |
| IF (Context = “Serving”) THEN (needTask (Take_Order, Food_Serve, Check_out)) | IF (Task = “Take_Order”) THEN(needAction (Give_Menu, Help_order)) |

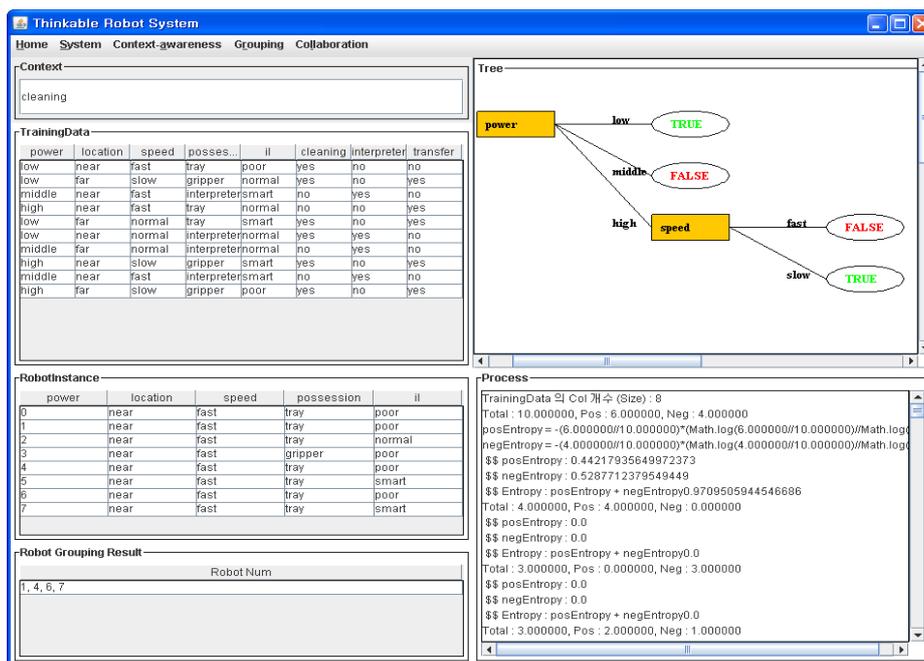


Fig. 6. Simulated Robot Grouping

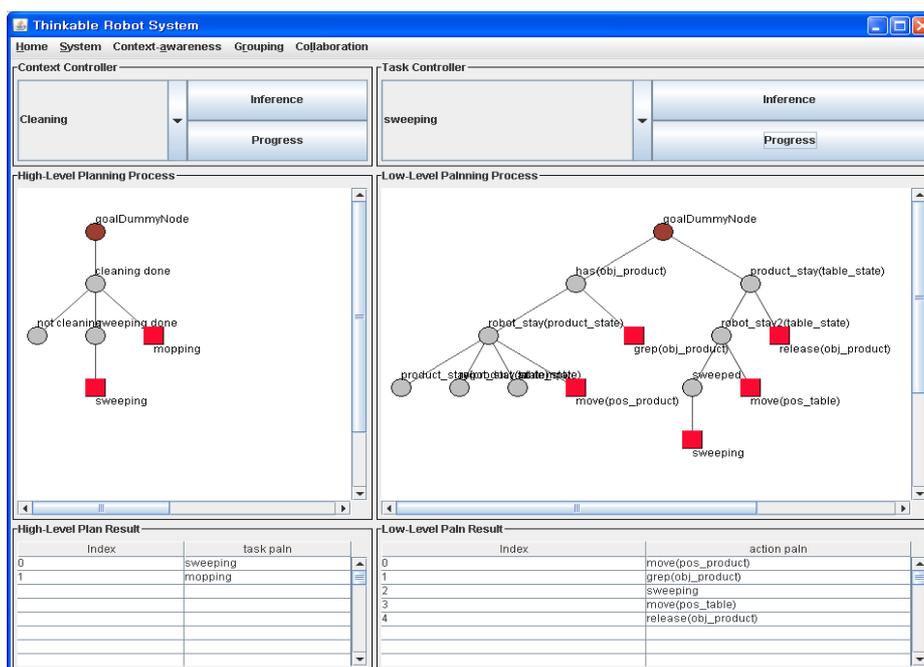


Fig. 7. Multi-level Task Planning

2) Multi-level Planning

The Fig. 7 shows two-level planning mechanism. On the left-hand side window shows the high-level planning process, which generates a set of tasks at the leaf nodes in square. On the right-hand side of window shows the entire process of action planning, which generates a set of actions as a result. The nodes in square are the actions that the robot must carry out.

V. CONCLUSION

In this paper, we have described a design and development of social robot framework for providing intelligent service in the context awareness environment. Some of the significances of our research are as follows. First, we have successfully showed our approach with the simulated experimentation by generating several robot instances. The heterogeneous robot instances can be generated by the user through the user interface windows, as many as possible.

Second, the robots are grouped by the generated decision classifier based on the entropy metric of the information theory. This kind of systematic approach will provide more reliable robot grouping method for the robot collaboration. Third, the selected context will generate a set of tasks, which will be allocated onto a group of robots based on the task allocation planning mechanism. The high-level task planning is done on the server side, and detailed action planning is done on the client side (robot).

Our approach may provide some useful solutions for the intelligent service oriented applications, which require multiple robot instances. Our immediate future work is to complete development of the context-awareness layer to use the sensed raw data from the computing environment. Therefore, when we complete the development of the entire framework, we are going to setup experimentations with several humanoid robots equipped with sensors and network communications.

REFERENCES

- [1] M. Baldauf, and S. Dustdar, "A Survey on Context-aware systems", *International Journal of Adhoc and Ubiquitous Computing*, Vol. 2, pp263-277, 2004.
- [2] G. K. Mostefaoui, J. Pasquier-Rocha, and P. Brezillon, "Context-Aware Computing: A Guide for the Pervasive Computing Community", *Proceedings of the IEEE/ACS International Conference on Pervasive Services (ICPS'04)*, vol. 19, issue 23, pp39-48, 2004.
- [3] G. Abowd, and A. Dey, "Towards a Better Understanding of Context and Context-Awareness", In: *LNCS1707*, pp304-307, 1999.
- [4] H. Chen, "An Intelligent Broker Architecture for Pervasive Context-Aware Systems", PhD Thesis, University of Maryland, 2004.
- [5] T. Gu, H.K. Pung, and D.Q., Zhang, "A Service-oriented middleware for building context-aware services", *Journal of Network and Computer Applications*, vol.28, pp1-18, 2005.
- [6] D. Salber, A. Dey, and G. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications", *Proceedings of CHI'99*, pp434-441, 1999.
- [7] X.H. Wang, T. Gu, D.Q. Zhang, and H.K. Pung, "Ontology Based Context Modeling and Reasoning using OWL", *Proceedings of 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops, PERCOMW'04*, pp18-22, 2004.
- [8] D. Rodic, and A.P. Engelbrecht, "Social network as a Coordination Techniques for Multi-Robot Systems", *International Conference on System Design and Applications*, Springer, Berlin, pp.503-513, 2003.
- [9] T.H. Labella, M. Dorigo, and J-L. Deneubourg, "Self-Organized Task Allocation in a Group of Robots", *Technical Report No. TR/IRIDIA/2004-6*, Universite Libre de Bruxelles, 2004.
- [10] M. Mataric, S. Sukhatme, and E. Ostergaard, "Multi-robot task allocation in Uncertain Environment", *Autonomous Robots*, vol.14, pp255-263, 2003.
- [11] M. Sterns, and N. Windelinckx, "Combining Planning with Reinforcement Learning for Multi-robot Task Allocation", *Proceedings of adaptive agents and MASII, LNAI 3394*, pp260-274, 2006.
- [12] R. Alami, A. Clodic, V Montreuil, E. Akin, and R. Chatila, "Task planning for Human-Robot Interaction," *Proceedings of the 2005 joint conference on Smart Object and ambient intelligence*, pp81-85. 2005.
- [13] B. Gerkey, and M. Mataric, "Principled Communication for Dynamic Multi-robot Task Allocation", *Experimental Robotics Vol. II*, pp353-362, 2000.
- [14] C.E. Shannon, *The Mathematical Theory of Communication*, University of Illinois Press, 1949.
- [15] M.A. Goodrich, E.R. Boer, J.W. Crandall, R.W. Ricks, and M.L. Quigley, "Behavioral Entropy in Human-Robot Interaction", In: *Brigham University, Technical Report ADA446467*, 2004.
- [16] T. Balch, "Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity", *Autonomous Robots*, vol. 8, pp209-237, 2000.