

# Chapter 2

## A Framework for Collaborative Aspects of Intelligent Service Robot

Joohee Suh and Chong-woo Woo

**Abstract** Intelligent service robot is becoming one of the most interesting issues in the recent Robot research. The service robot monitors its surroundings, and provides a service to meet a user's goal. The service often becomes too complex that one single robot may not handle efficiently. In other words, a group of robots may be needed to accomplish given task(s) by collaborating each other. We can define this activity as a robot grouping, and we need to study further to make better group(s) by considering their characteristics of the each robot. But, it is difficult and no formal methods to make such a specific group from the many heterogeneous robots that are different in their functions and structures. This paper describes an intelligent service robot framework that outlines a multi-layer structure, which is suitable to make a particular group of robots to solve given task by collaborating with other robots. Simulated experimentation for grouping from the generated several heterogeneous is done by utilizing *Entropy* algorithm. And the collaboration among the robots is done by the multi-level task planning mechanism.

### 1 Introduction

Ubiquitous computing [1] defines that various computing objects are connected through the network, so that the system automatically provides services in anytime in any place. An intelligent robot is an autonomous and dynamic object in this ubiquitous environment, and it becomes one of the most interesting issues in this area. It interacts with various surrounding computing devices, recognizes context,

---

J. Suh (✉)

Korea School of Computer Science, Kookmin University, 861-1 Jeongneung-Dong, Seongbuk-Gu, Seoul

e-mail: crazyDMP@gmail.com

and provides appropriate services to human and the environment [2]. The definition of the context [3] is any entity that affects interactions among the computing objects in this environment. For instance, *user*, *physical location*, *robots* could be such entities. Therefore, information described for the characteristics of such entities are defined as a context, and recognizing a situation from the context is the main focus of the context awareness system [4]. Essentially, the robot is required to carry on two main issues in this study; understanding the context and carrying out the context.

First, understanding a context can be done in many different ways, but the general procedure includes that system perceives raw environmental information through the physical sensors. And then context modeling and reasoning steps follows right after preprocessing the raw data. The final result from this procedure is a context. The approaches on this procedure are vary, and it needs to be further studied in detail to make the system more efficient, and there are large amount of studies [2-4] are being reported recently. The second issue is that the robot has to carry out the generated context to meet the user's goal. The context can be further divided into some tasks, which often becomes too complex that one single robot may not handle in this environment. In other words, a group of robots may be needed to accomplish given task(s) by collaborating each other. We can define this activity as a robot grouping, and we need to study further to make better group(s) by considering their characteristics of the robot and the task.

In this paper, we are describing a 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 understanding part of the system is designed, but will be discussed in the next study with comprehensive ontological knowledge representations along with context modeling and context reasoning mechanisms.

## 2 Related Works

### 2.1 Context-Awareness Systems

With the increase of mobile computing devices, the ubiquitous and pervasive computing is getting popular recently. 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 [5] (Context Broker Architecture), SOCAM [6] (Service Oriented Context-Aware Middleware), and Context-Toolkit [7]. The CoBrA and SOCAM used ontological model aiming for the benefits of easy sharing, reusing, and reasoning

of context information. But, first, these systems were not flexible enough to extend to other domains when the system needs to expand with other devices or service with a new domain. And second, they were also limited in formalized and shared expression of the context, which is needed when the system interoperate or transplant with other systems. Therefore, the ontology becomes one of the most popular solutions 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, it lacks of the standardized representation for the context, and also the interoperability between heterogeneous systems.

**CoBrA:** This system is developed based on the ontology, so that 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 adapted merits of the previously studied systems, and designed a framework that can overcome the above limitations, such as, the limited standardized representation or extensibility. For instance, we have adapted context-awareness layer by adapting the CONCON model [8] to provide extensibility of the ontological representation.

## 2.2 *Robot Grouping and Collaboration*

First of all, our research issue focused on the collaboration among the robots, which needs to form 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 [9] 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. [10], 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. [11] had an experimentation comparing between a simulated data with physical mobile robot experiment. The result showed that there is no single strategy that produces

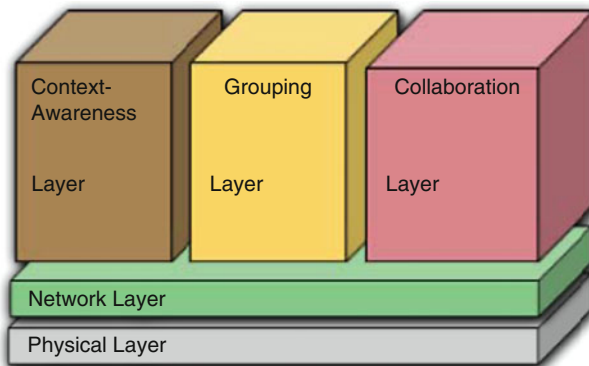
best performance in all cases. And other approaches are the multi-robot task allocation by planning algorithm [12–14].

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 [15] could be a good alternative compare to the other informal approaches. Goodrich argues that the behavioral entropy can predict human workload or measure of human performance in human robot interaction [16] (HRI) domain. Balch [17] 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.

### 3 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 the next Fig. 2.

- The context-awareness layer generates a context from the raw information, and then does modeling and reasoning in order to aware of the context.
- The grouping layer creates decision classifier based on the entropy mechanism, and makes a necessary group.



**Fig. 1** General structure of social robot

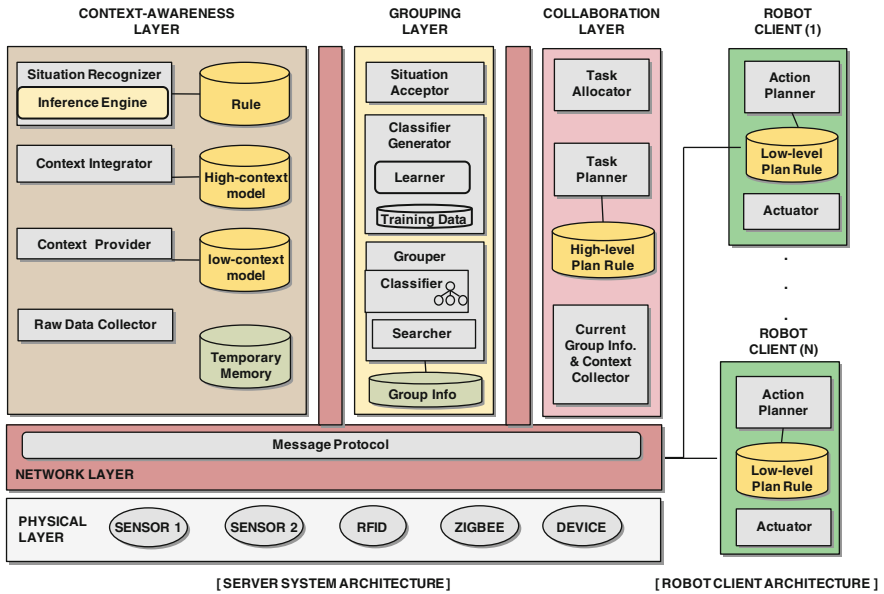


Fig. 2 Structure of the service robot in context awareness environment

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

### 3.1 Context-Awareness Layer

The context awareness layer receives raw data from surrounding computing devices including RFID, Zigbee, and so on. Then it transforms the raw data into a meaningful semantic data by going through some preprocessing steps, and finally it generates a situation. This work can be done by the next sub modules as follows.

**Raw Data Collector:** It simply 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* as a preprocessing step according to low context model. The low context model means that the raw data is formalized but it is not semantic data.

**Context Integrator:** It receives standardized context from the *context provider* and generates inference level context through the high-level context modeling. The high level model supports converting the formalized context into a semantic context.

**Situation Recognizer:** The context awareness process goes through the above modules in sequence, and generates *situation(s)* by using rule-based inference engine in this sub module. This situation is delivered to the *grouping layer*.

## 3.2 Grouping Layer

When the situation generation is done by the situation acceptor, then this situation is delivered to the *grouping layer* to make a group. The grouping layer first receives information about which robot is connected to the server, and stores this information into *Group info* database. Because currently connected robots through the network can be the candidates for making a group. The grouping layer consists of three sub modules, *Situation Acceptor*, *Classifier Generator*, and *Grouper*, and the details of their work are as follows.

**Situation Acceptor:** It simply receives information regarding a *situation*, from the Context awareness layer, and requests to the *Classifier Generator* to begin grouping for this given situation.

**Classifier Generator:** It generates a *classifier* (i.e. decision tree) to make a group for a given specific situation. And also we need to have a set of predefined training data representing some characteristics of various kinds of robots. In this study, we generated the *classifier* based on ID3 decision tree algorithm.

**Grouper:** The Grouper has the next two sub-modules; the *searcher* requests instance information from the connected each robot through the network layer. In other words, after the request for the instance information is acquired, such as ‘cleaning’, the grouper makes a group by using the *classifier* that generated from the *Classifier Generator* module. The generated group information is stored in the *group info* repository, and will be used for collaboration in the collaboration layer later on.

### 3.2.1 Context Information for the Robot

For this experiment, we can set up several virtual situations, such as ‘cleaning’ situation, ‘delivery’ situation, ‘conversation’ situation, and so on. The Grouping layer receives one from the situations, and start making a group that is appropriate for the service.

The Fig. 3 is the user interface for entering robot attributes and a situation. From this interface, we can enter five attributes interactively, such as ‘power’, ‘location’, ‘speed’, ‘possession’, ‘IL’, and a situation, such as ‘cleaning’. By using this interface, we can create robot instances as many as we want arbitrarily and they represent heterogeneous robots. And also we can set up a situation by just selecting from the window. This means that each robot has different characteristics and good for a certain work. For instance, we can set up a robot instance

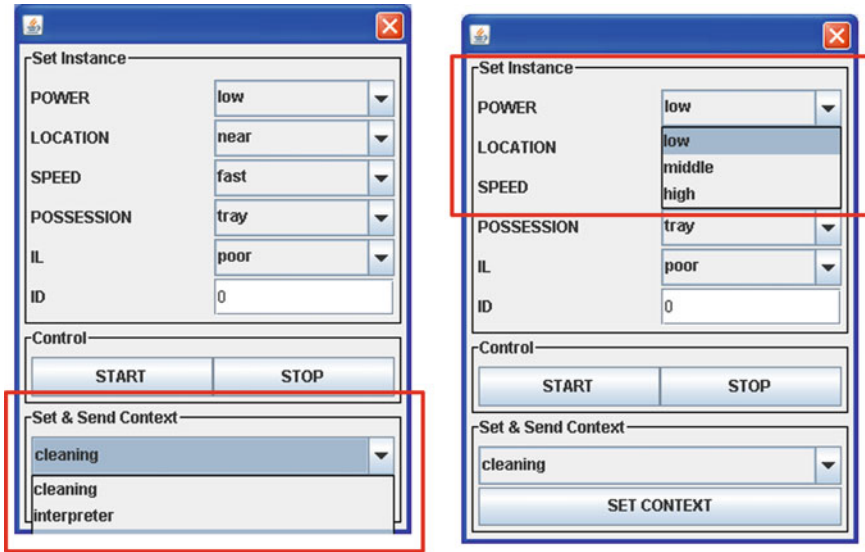


Fig. 3 User Interface for entering robot attributes and situation

Table 1 Training data to create a decision tree

Index	Power	Location	Speed	Possession	IL	Cleaning
1	Low	Near	Fast	Tray	Poor	Yes
2	Low	Far	Low	Gripper	Normal	Yes
3	Middle	Near	Fast	Interpreter	Smart	No
4	High	Near	Far	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	Far	Interpreter	Smart	No
10	High	Far	Slow	Gripper	Poor	Yes

good for a ‘cleaning’ job as follows. If the robot has low power, location is near, speed is low, possesses a tray, and so on, then we can consider this robot is good for a cleaning situation. Similarly, we can create several robot instances for our experimentation.

### 3.2.2 Training Data

Each robot’s attributes are described as in the Table 1, which shows that each robot has different characteristics. For instance, ‘power, and speed’ means the current robot’s basic characteristics, ‘location’ is the robot’s location to perform the given

context, ‘possession’ means a tool that robot can handle, and ‘IL’ means the robot’s capability of language interpretation. Since we are not using sensed data from computing devices for this simulation, we can generate robot instances through the user interface arbitrarily, as many as possible. Table 1 is a set of training data that shows ten robot instances for ‘cleaning’ situation.

### 3.2.3 Decision Tree Generation

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

$$Entropy(S) = -P(+)\log_2 P(+)-P(-)\log_2 P(-) \quad (1)$$

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

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

## 3.3 Collaboration Layer

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

**Group Info. and Context Collector:** It collects information of selected situation and information for grouped robots from the *Grouping Layer*.

**Task Planner:** It generates a set of tasks using high-level planning rules (a global plan) on the server side. 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.

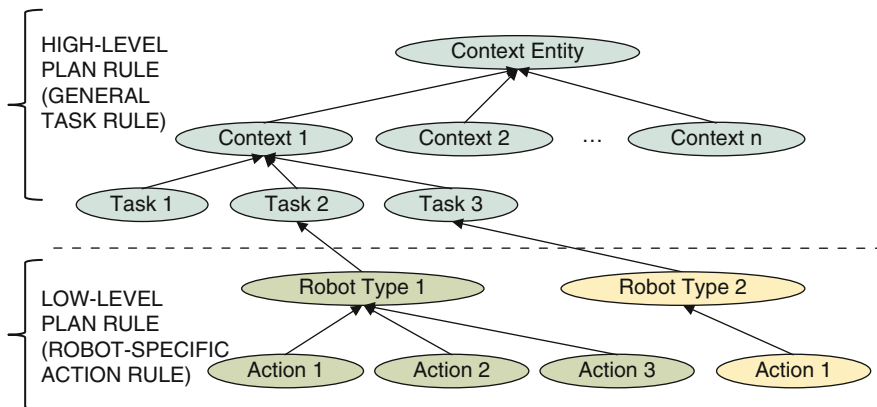
**Action Planner:** Generated tasks by the *task planner* are delivered to the client (robot), and further refined into a set of actions by the action planner.

### 3.3.1 Multi-level Task Planning

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

- The task planner gets the situation(s), and generates a set of tasks based on the high-level planning rules.





**Fig. 4** Multi-level task planning mechanism

- Then the system allocates the tasks to the appropriate robot who can handle the specific task.
- When the task allocation is done, then each assigned robot activates the action planner to generate a set of actions.

## 4 Simulated Experimentation

The overall experimentation is divided into two parts, robot grouping and robot collaboration. Robot grouping is done by generating classifier using the Entropy metric, and the collaboration is done by task planning algorithm.

### 4.1 Robot Grouping

In this study, the robot grouping simulation experiment begins with generating virtual robot instances and a situation through the user interface as in the Fig. 3. We can set up characteristics of each robot by selecting five attributes and also can set up a virtual situation through the interface. When all the selection is done, then we can send the information to the server using the start/stop button in the interface.

Figure 5 shows the snapshot of the implemented simulation result for grouping. We designed the implementation result as six sub windows, and the function of each window is explained as follows.

- ‘Context’ window: It shows the selected virtual context, such as, ‘cleaning’.
- ‘Training Data’ window: It shows ten training data for the selected situation.

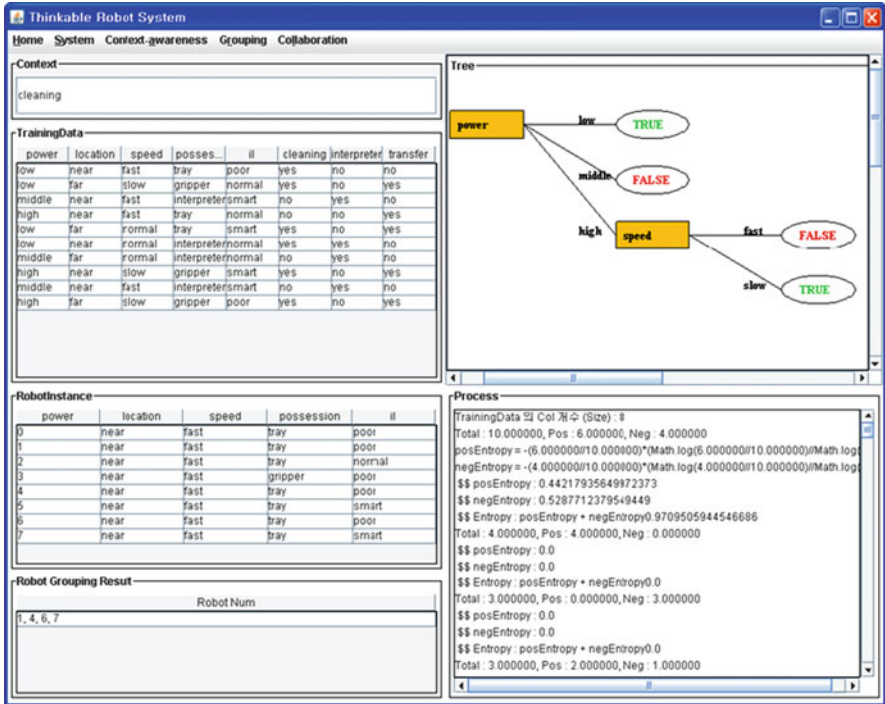


Fig. 5 Snapshot of simulated robot grouping

- ‘Robot Instance’ window: It shows activated instances.
- ‘Robot Grouping Result’ window: It shows grouped robots as instance numbers.
- ‘Tree’ windows: It shows the generated decision tree for the selected context.
- ‘Process’ window: It shows the entropy computation process for the generated decision tree.

The entire grouping process can be summarized as follows:

- Create arbitrary number of robot instances through the user interface (see Fig. 3), and this information is sent to server.
- Also, information of a ‘virtual situation’ is sent to server.
- Then, the server creates decision tree using the information of the robot instances, and decide a group of robots for the specific situation (e.g. cleaning).
- 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. 5, we can see that the robot instance number 1, 4, 6 and 7 are grouped together to perform the task, ‘cleaning’.

## 4.2 Robot Collaboration

Although there are many different approaches of collaborations in robot studies, we consider the collaboration as sharing a task among a group of robots. For instance, a generated task can be further refined into a set of subtask, which will be carried out by each individual robot. The system generates a set of tasks or subtasks by the task planning mechanism, which will be described in the below.

### 4.2.1 Task Planning Rules

The task planning rules for collaboration layer is divided into two 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.

### 4.2.2 Multi-level Planning

Our system divides the planning mechanism in two levels, a high-level planning and low-level planning. It is a kind of hierarchical planning mechanism that is efficient enough to make the planning mechanism as simple as possible. The high-level planner generates a set of subtasks to accomplish a given context, and saves them in a stack that will be taken one by one by the low-level planner. When the high-level planning is done, the low-level planner gets activated with each one of the subtasks. A single subtask becomes a goal to accomplish in the low-level planning process, and it generates a set of actions as a result.

**Table 2** Task planning rules

High-level planning rules	Low-level planning rules
If (Context = ‘Cleaning’) then (NeedTask Sweeping, Mopping)	If (Task = ‘Sweeping’) then (NeedAction (Move, Lift, Sweeping, Release))
If (Context = ‘Serving’) then (NeedTask (Take_order, Food_service, Check_out)) . . .	If (Task = ‘Take_order’) then (NeedAction (Give_Menu, Help_order)) . . .

## 5 Conclusion

In this paper, we have described a development of intelligent robot framework for providing intelligent service in the context awareness environment. The significances of our research could be as follows.

First, we have designed intelligent service robot framework and successfully carried out simulated experimentation by generating several robot instances. The several 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 decision classifier based on the entropy metric of the information theory. This approach will provide more reliable and systematic methods for the robot grouping. Third, in this study, we consider the collaboration as follows. Once a task is generated, then it gets refined into a set of subtasks, which will be assigned to each individual robot. If each robot accomplishes its own subtask, then the original task is being done as a result. We do not have any interaction among the robot at the moment. Fourth, the generation of task is being done by the multi-level 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 next work is to complete the development of context-awareness layer using sensed raw data from the external physical computing devices. 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. Also, more detailed strategy and also interaction for collaboration among each individual robot is needed.

**Acknowledgement** This work was supported by the Seoul R&BD program (10848) of Korea.

## References

1. M. Weiser. The computer for the twenty-first century. *Sci. Am.* **265**(3), 94–104 (1991)
2. G.K. Mostefaoui, J. Pasquier-Rocha, P. Brezillon, Context-aware computing: a guide for the pervasive computing community. *Proc. IEEE/ACS Int. Conf. Pervasive Serv. (ICPS'04)* **19**(23), 39–48 (2004)
3. A.K. Dey, G.D. Abowd, Towards a better understanding of context and context-awareness. *GVU Technical Report GIT-GVU-99-22*, Georgia Institute of Technology, 1999
4. M. Baldauf, S. Dustdar, A Survey on context-aware systems. *Int. J. Adhoc Ubiquitous Comput.* **2**, 263–277 (2004)
5. H. Chen, An intelligent broker architecture for pervasive context-aware systems. Ph.D. thesis, University of Maryland, 2004
6. T. Gu, H.K. Pung, D.Q. Zhang, A Service-oriented middleware for building context-aware services. *J. Netw. Comput. Appl.* **28**, 1–18 (2005)
7. D. Salber, A. Dey, G. Abowd, The context toolkit: aiding the development of context-enabled applications, in *Proceedings of CHI'99*, Pittsburgh, PA, 1999, pp. 434–441

8. X.H. Wang, T. Gu, D.Q. Zhang, H.K. Pung, Ontology based context modeling and reasoning using owl, in *Proceedings of 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops, PERCOMW'04*, Orlando, Florida, USA, 2004, pp. 18–22
9. D. Rodic, A.P. Engelbrecht, Social network as a coordination techniques for multi-robot systems. International conference on system design and applications, Springer, Berlin, 2003, pp. 503–513
10. T.H. Labella, M. Dorigo, J-L. Deneubourg, Self-organized task allocation in a group of robots. Technical Report No. TR/IRIDIA/2004–6, Universite Libre de Bruxelles, 2004
11. M. Mataric, S. Sukhatme, E. Ostergaard, Multi-robot task allocation in Uncertain Environment. *Auton. Robots* **14**, 255–263 (2003)
12. M. Sterns, N. Windelinckx Combining planning with reinforcement learning for multi-robot task allocation. in *Proceedings of adaptive agents and MASII*, LNAI 3394, 2006, pp. 260–274
13. R. Alami, A. Clodic, V. Montreuil, E. Akin, R. Chatila, Task planning for human-robot interaction, in *Proceedings of the 2005 joint conference on Smart Object and ambient intelligence*, 2005, pp. 81–85
14. B. Gerkey, and M. Mataric, Principled communication for dynamic multi-robot task allocation. *Exp. Robotics* **II**, 353–362 (2000)
15. C.E. Shannon, *The Mathematical Theory of Communication* (University of Illinois Press, Illinois, 1949)
16. M.A. Goodrich, E.R. Boer, J.W. Crandall, R.W. Ricks, M.L. Quigley, Behavioral entropy in human-robot interaction, Brigham University, Technical Report ADA446467, 2004
17. T. Balch, Hierarchic social entropy: an information theoretic measure of robot group diversity. *Auton. Robots* **8**, 09–237 (2000)