A Platform System for Developing a Collaborative Mutually Adaptive Agent

Yong Xu[1], Yoshimasa Ohmoto[1], Kazuhiro Ueda[2], Takanori Komatsu[3], Takeshi Okadome[4], Koji Kamei[4], Shogo Okada[1], Yasuyuki Sumi[1], and Toyoaki Nishida[1]

[1] Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
[2] Department of System Sciences, the University of Tokyo
[3] International Young Researcher Empowement Center, Shinshu University
[4] Innovative Communication Laboratory, NTT Communication Science Laboratories

xuyong@ii.ist.i.kyoto-u.ac.jp,
{ohmoto,okada_s,sumi,nishida}@i.kyoto-u.ac.jp,
ueda@gregorio.c.u-tokyo.ac.jp, tkomat@shinshu-u.ac.jp,
houmi@idea.brl.ntt.co.jp, kamei@cslab.kecl.ntt.co.jp

Abstract. The characteristic task of service robots that can interact with humans is to achieve human-robot collaboration. Mutual adaptation is considered to be an important characteristic of robots, required for carrying out such collaborative tasks. Here, we introduce the concept of mutual adaptation, propose a learning model, and describe an experimental task to explain the above concept. A waiter robot performs a collaborative task using a platform system, which is developed by a constructive approach. The interactive and manual modes of this system are compared by performing a preliminary experiment to evaluate the effectiveness of the robot's autonomous function. The results indicate that the robot's autonomous function works well when operated in the interactive mode under short time or slow speed conditions.

Keywords: Human Robot, Intelligent Interfaces, Mutual Adaptation.

1 Introduction

The characteristic task of service robots that can interact with human users is to achieve human-robot collaboration. At present, research is being conducted on developing an intelligent robot (agent) that is capable of receiving and learning undefined instructions from humans. Thomaz et al. [1] developed an interactive computer game system known as Sophie's Kitchen, in order to enable humans to teach a robot and simultaneously improve the robot's learning behavior. From the results of their research, it was found that humans dynamically change their behavior as they develop a mental model of a robotic learner. However, their study mainly focused on a reinforcement learning task where humans know the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 576–585, 2009.

goal of a task which is not known by the robot; in such a situation, humans often take initiative. Yamada et al. [2] proposed the concept of "adaptation gap." They argued that there often exists an adaptation gap between the expected ability and real ability when human users and robots try to build an internal prediction model for each other. The results of their study suggest that in the case of collaborative agents, especially agents that are required to interact with humans, it is very difficult to develop a completely autonomous agent that can accomplish collaborative tasks by adapting to humans and simultaneously ensure that humans can easily teach the robotic learner. Authors argue that mutual adaptation is an important characteristic of collaborative robots. Service robots apparently differ from traditional industry robots because they are generally required to interact with human users in an uncontrolled environment. Among various types of tasks of service robots, the task of a waiter robot is a typical task.

2 Concept of Mutual Adaptation

2.1 Definition and Hierarchical Structure

First, it is assumed that there are two agents A and B with different abilities; A and B perform the same collaborative task. In this task, A and B can only share partial information with each other. In order to achieve a common purpose, each agent is required to build a model for the other, develop a communication protocol that can be used to interpret information from the other, and draw inferences about the other depending on various situations. In general, the agents have to complete the task by gradually adapting to each other. In this paper, the above-mentioned phenomenon is termed *mutual adaptation*.

Mutual adaptation is observed in human-human communication. Xu et al. [3] performed an experiment, the result of which indicates that mutual adaptation has a hierarchical structure. Xu et al. [4] developed a two-layered model, which indicates that mutually adaptive behaviors can take advantage of at least two levels of mutual adaptation protocols. In the lower level, basic protocols are developed using fundamental elements. In the upper level, abstract protocols are developed by combining elements from the lower level or creating new elements.

2.2 Preconditions and Occurrence Conditions

In order to design an experiment to invoke mutual adaptation, it is necessary to define the preconditions of mutual adaptation, which are summarized as follows.

– Asymmetric ability: The abilities of A and B are said to be asymmetric if the instructor agent is a human and the learner agent is a robot. Both the agents require the following abilities to perform the task in this study:

$$A_{human} = A_{common} + \alpha \tag{1}$$

$$A_{robot} = A_{common}, \tag{2}$$

where A_{human} represents the ability of the human, A_{robot} represents the ability of the robot, and A_{common} represents the abilities common to both the human and the robot.
(1)Abilities common to both the human and robot
- Ability to understand rewards received from the partner
- Ability to take optimal actions
- Ability to take initiative (autonomous actions)

(2)Ability of α of the human instructor
- Ability to develop new concepts
- Ability to propose new protocols
- Ability to build a model for the entire human-robot collaborative system

Although the robot developed by present technologies may lack the ability to mutually adapt, it is still necessary to clarify how humans with the above-mentioned abilities are capable of mutually adapting.

- Asymmetric information: When A and B obtain only asymmetric information, it implies that each agent does not have complete information that is necessary for completing the collaborative task.
- Sharable entire evaluation score: It is necessary to provide an entire evaluation score that both A and B can obtain simultaneously. In order to get a high score, it is necessary to design a suitable task, which cannot be accomplished without successfully sharing information, and developing and exchanging protocols (communication) between the two agents.
- Transition initiative: In some situations where A acquires sufficient information, A may always take initiative and instruct B to follow its instructions. However, in some other situations where B has some information that A does not have, or B knows that by performing an autonomous action, a high score can be achieve, B may perform the task on its own, so that the initiative is transferred from A to B.

In general, the sufficient and necessary conditions can be expressed as $X \rightarrow Y$, which implies that X is the sufficient condition of Y, and Y is the necessary condition of X.

Sufficient condition: If the following conditions are satisfied, there is a high possibility for the occurrence of mutual adaptation.

- The human instructor is aware of the merit of robot's autonomous action.
- The human instructor changes his/her strategy or instruction approach to adapt to the robot.

Necessary condition: If a mutual adaptation phenomenon occurs, there is a high possibility for the occurrence of the following:

- Each agent understands the behaviors (instructions or actions) of the other, irrespective of whether the partner was aware of it or not.
- The information shared is asymmetric.
- Both the agents take initiative.
- Common protocols are developed.

On the basis of the above-mentioned definition and occurrence conditions of mutual adaptation, it is necessary to verify them by performing a specific human-robot collaborative task.

3 Experiment to Clarify Mutual Adaptation

3.1 Objective

The objective of this study is to explain the concept of mutual adaptation by considering a training task of a waiter robot; in this task, a manager (human instructor) trains a waiter (robot) in a restaurant. It is essential that the degree of satisfaction of the customers and the profit of the restaurant are maximized simultaneously. Here, the degree of satisfaction of the customers is reflected by the amount of tip given to the waiter robot.

The objective of this study is to enable instruction-based learning, but not model-based learning. This is because the meaning of the same instruction can be interpreted differently, depending on different situations. This study focuses on not only learning new environments, but also learning new possible instructions. To obtain an ideal result, human users are expected to be aware of the merit of autonomous actions performed by the robot; on the basis of the obtained result, humans give the same instructions implying different meanings under different situations. On the other hand, robots are expected to respond to instructions properly by developing common protocols, adapting to changes, and demonstrating their capabilities in learning human models, so that mutual adaptation can be invoked.

3.2 Task

The waiter robot performs its task by using a software simulation system. The task is to be performed by a manager (human instructor), waiter (software robot), and some customers (generated and controlled by an automatic program). The table layout is shown in Fig. 1. Each table has up to the following five states: “vacant table,” “waiting for guide (only available for entrance),” “waiting for order,” “eating,” and “clearing table.”

3.3 Model

There are two types of instructions that can be given by the manager, namely, action instruction and evaluation instruction. The manager can decide the target table for the robot by referring to the current position of the robot; he can provide the seat number of the destination to the robot in the form of instructions. The objective of the task is to maximize the sales of the restaurant (i.e., minimize the total stay time of all the customers) and simultaneously maximize the scores of the entire evaluation function (i.e., maximize tips received by the robot and the sales of the restaurant) by minimizing the movement time of the robot. On

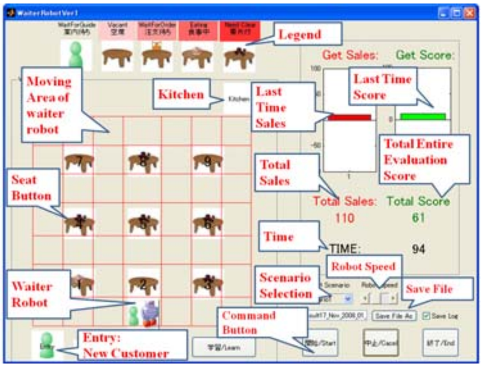

Fig. 1. Graphical User Interface of the waiter robot simulation system

the other hand, the robot simultaneously aims to maximize the tips received from the customers and the sales of the restaurant by minimizing the customers' waiting time before they are guided to a seat or before the robot takes their order.

In principle, the waiter robot has two types of choices i.e., it can move by following the manager's instruction, or it can move autonomously by complying with its build-in rules or by using previous knowledge. For example, the robot may move towards a customer who is sitting at the table nearest to its current position and waiting to place his order, or it may move along the shortest route to provide services to two or more customers who are waiting to be served. In general, the robot always tries to move along the best route, in order to receive maximum tips and reduce the waiting time of the customers'. The robot needs to strike a balance between maximizing tips and minimizing waiting time. In order to achieve the former, the robot moves along a complex route to provide services to the maximum possible customers. However, in order to achieve the latter, the robot moves along a simple route to a customer who has been waiting to be served a long time. A customer who has been waiting for a very long time may leave the restaurant, causing a loss of sales and revenue; therefore, it is necessary for the robot to clear the tables immediately after the customers leave, so that there is a sufficient number of available vacant tables for new customers.

There are three types of rewards, namely, sales, tips, and scores. The amount of restaurant's sales is a type of reward that the manager aims to maximize. This amount relates to the number of customers who place orders and do not leave. The amount of tips received by the robot can only be known by it. The robot always tries to maximize the tips it received. Changing trends in scores are considered to be a common reward that is shared by both the manager and the waiter robot.

The relationship between the three evaluation functions is shown in Fig. 2. The first function of tips is only applicable to the robot; this function enables the robot to choose its default action depending on the expected values of the tip. On the other hand, the human instructor, who plays the role of a manager,

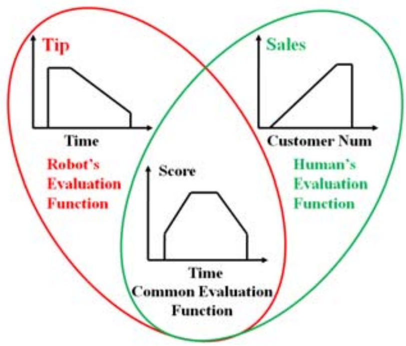

Fig. 2. Evaluation Function

has no knowledge of the amount of tip received; therefore the manager prefers to instruct the robot to serve the table nearest to its current position. The entire evaluation function is assumed to be shared by the robot and the manager.

The learning function will be implemented by using the model as shown in Fig. 3. This model mainly consists of the following four modules: human instruction evaluator (HIE), robot action evaluator (RAE) , robot action generator (RAG) , and robot action decision-maker (RAD). The main functions of the HIE and RAE modules are to evaluate the manager's instruction and robot's action; the score changes on the basis of the result of the evaluation. The RAD module compares the results obtained by the above two modules and decides the type of action that the robot is required to perform; the actions are defined in the RAE and RAG modules. The results of the real robot action are provided to both the RAE and RAG modules, so that they can be used as references for learning in future. The score calculation function is used to generate the score for the simulation system; however, this function is not known by the robot. If the robot has knowledge about the score, it can always take optimal action without receiving any instructions from the manager.

In order to develop such an experimental system, which is expected to be able to invoke mutual adaptation, it is necessary to develop a platform system where the autonomous function of the robot can work well to generate sufficient scores without receiving any instructions from the manager. In order to evaluate the effectiveness of the robot's autonomous function, a preliminary experiment is performed. The details of this experiment will be described in the next section.

3.4 Method

A waiter robot simulator software system with a graphical user interface was developed, as shown in Fig. 1. Human participants can instruct the robot by pressing buttons. They are informed in advance that the robot can perform some autonomous actions based on its own decisions. The goal of the task is to maximize the sales and the scores within a limited time. First, the platform

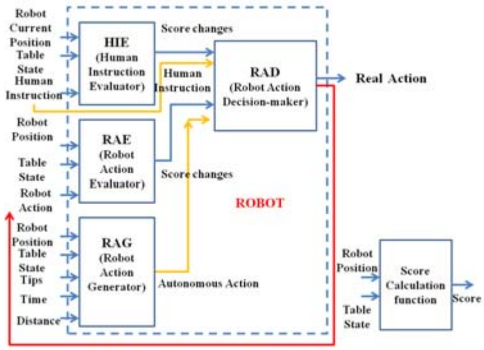

Fig. 3. Learning Model

system with an autonomous robot capable of interactive responses needs to be tested. If the autonomous function is very poor, it may become necessary for the manager to intervene regularly. On the other hand, if the function is very perfect, the manager may issue instructions and then simply observe the robot performing its tasks without giving any further instructions. Both these cases are unsuitable for invoking mutual adaptation.

A typical scenario is as follows. When a new customer enters the restaurant, if a seat is vacant and the robot is idle, it should move to the entrance and guide the customer to the vacant seat. The waiter robot is assumed to have three states: "normal," "guiding customer," and "carrying dish." When it is guiding a customer, its state will change to "guiding customer" automatically, and until the customer is seated, the manager's instructions will be ignored. If there is neither an available vacant seat nor a seated customer waiting to place an order, the robot needs to find a "need to clear" table, move there, clear the table, and change its state to "carrying dish" while carrying dishes to the kitchen. However, if any customer is waiting to place an order, the robot should first take the order by obeying its tip-first law. After the customer has been seated at a vacant seat, the table state will be changed to "waiting for order" state. It will be kept unavailable for placing orders for a specific period. In order to prevent customers from leaving the restaurant because they have waited too long time, which in turn reduces the sales and scores, the robot always tries to place an order as soon as possible when operating in the interactive mode. When the robot finishes placing an order, it receives tips, and the restaurant's sales and scores are updated simultaneously. All these data are recorded into log files.

In order to verify the effectiveness of the robot's autonomous function, an interactive mode is implemented in addition to the manual mode. Since the goal of this system is to invoke mutual adaptation, it is reasonable to expect the robot to obtain a sufficiently high score through its autonomous function. It is expected that the manager's method of instructing will be affected by the robot's autonomous function. This in turn will enable the robot to learn and improve its performance and adapt to the manager by trying various actions. Therefore, it is desirable to invoke mutual adaptation.

Table 1. Comparison Results for Manual & Interactive Mode with Statistical Value

TrialNo	RunningMode	Speed	Time(s)	N	GetScoreMean(S.D.)	GetSalesMean(S.D.)
1	manual	fast	60	11	76.69(14.10)	159.72(29.44)
1	interactive	fast	60	11	84.85(16.88)	178.09(35.22)
2	manual	fast	60	9	73.87(6.93)	139.67(17.31)
2	interactive	fast	60	9	84.49(3.69)	166.22(9.23)
3	manual	fast	300	9	273.80(37.50)	562.22(80.69)
3	interactive	fast	300	9	257.76(23.96)	535.44(50.37)
1	manual	slow	60	9	39.9(18.36)	82.33(38.58)
1	interactive	slow	60	9	56.13(13.49)	117.0(27.97)
2	manual	slow	60	10	38.08(13.12)	82.7(22.17)
2	interactive	slow	60	10	56.46(9.56)	123.4(15.06)
3	manual	slow	300	1	173.6	344.0
3	interactive	slow	300	1	183.0	375.5

3.5 Results and Discussions

A software platform for the waiter robot simulator system was developed, and among the 10 scenarios available, 3 were tested. Two modes of the robot, namely, the manual and interactive modes, were implemented in this system. In the manual mode, the waiter robot cannot move without receiving instructions from the manager. In contrast, in the interactive mode, the robot can move autonomously without requiring instructions from the manager or having to change its own actions to follow the manager's instructions. The robot can move at fast (one step per 0.5 s) or slow speeds (one step per 1 s). Two types of user interfaces, namely, a touch panel and a mouse, can be used to control the robot.

10 students (9 male and 1 female, age: 22-33 years, average age: 26) participated in this experiment.First, the participants were asked to complete the first trial using both the touch panel and the mouse at fast and slow speeds. Then, they were asked to complete the second trial using their preferred user interface at fast and slow speeds. Finally, they were asked to complete the last trial using their preferred user interface and speed. In each trial, the participants were asked to finish the task in both the manual and interactive modes. The results of all trials including the scores, sales, robot states, table states, and the manager's press-button actions were recorded in log files.

The results of the mean values of the scores and sales achieved by the participants during the experiment are listed in Table 1.

A series of two-sided paired-sample T-tests are applied to the experimental results to elucidate the differences between the manual and interactive modes. We consider the following null hypotheses for the three trials: there is no significant difference between the score and sales achieved by instructors when performing the task in the manual and interactive modes. The results of the T-test are listed in Table 2.

Table 2. Comparison Results for Manual & Interactive Mode Using Paired T-test

TrialNo	Speed	Time(s)	GetScore	GetSales
1	fast	60	m-mode<i-mode,t(10)=1.0 n.s.	m-mode<i-mode,t(10)=1.1 n.s.
2	fast	60	m-mode<i-mode,t(8)=3.6 **	m-mode<i-mode,t(8)=3.6 **
3	fast	300	m-mode>i-mode,t(8)=1.79 n.s.	m-mode>i-mode,t(8)=1.4 n.s.
1	slow	60	m-mode<i-mode,t(8)=4.396 **	m-mode<i-mode,t(8)=4.501 **
2	slow	60	m-mode<i-mode,t(9)=3.71 **	m-mode<i-mode,t(9)=4.63 **

(**:p< 0.01,*:p<0.05,+:p< 0.1,n.s.:p> 0.1,
m-mode:manaul mode,i-mode:interactive mode,df:degree of freedom)

With regard to the results, in the case of trial 1, the "score" and "sales" values differed significantly only under the slow speed condition, but not under the fast speed condition. In the case of trial 2, the two values differed significantly under both speed conditions. However, in the case of trial 3, no significant difference between the two values was observed.

The statistical analysis results indicate that the performance of the robot's autonomous function is better under short time trials or slow speed conditions than under long time trials or fast speed conditions. This implies that the robot's autonomous function works well in at least some situations.

On the basis of only the current results, it is difficult to confirm whether mutual adaptation is affected by the robot's autonomous function. Without considering the structure of the task, it may be difficult to understand why the performance of the interactive mode differed significantly from that of the manual mode in short time or slow speed trials, but not in long time or fast speed trials. We believe that in a collaborative task, it will be necessary for the human instructor to integrate his/her instruction model with the robot's autonomous functions to improve performance. In the interactive mode, some instructors issued few instructions and spent more time observing the robot's autonomous actions. This may imply that they tried to understand the intentions of the robot. In contrast, some instructors issued many instructions to the robot to make it follow their intentions. The current version of the system only allows for very simple autonomous functions, and therefore, it might not work effectively in long time trials. In order to improve the performance, it would be necessary for the robot to integrate the human instructor's instruction model with its own autonomous ability using some learning method. We believe that our mutual adaptation model is a solution to this problem. 8 out of 10 participants used both user interfaces (touch panel and mouse) in the trials; of these, only 3 preferred the touch panel. With regard to the robot's movement speed, 90% of the participants preferred the fast speed. Therefore, these parameters will be set as default parameters in future experiments.

A learning model is being developed based on the autonomous function of the robot in the platform system to enable it to learn from collaborative experience. Further, the participants' instruction behaviors will be analyzed to investigate how human instructors change their instruction methods. On the basis of these results, a typical nonparametric technique known as the PNN (*Probabilistic Neural Network*) model introduced in [5] will be used to implement a learning function for the robot. Since this system is implemented in the form of a game, Game theory [6] is expected to provide some theoretical references that will help formalize the problem. By combining game theory and machine learning, it may be possible to invoke multistage mutual adaptation.

4 Conclusion

This paper introduces the concept of mutual adaptation, proposes a learning model, and describes a task to explain the concept of mutual adaptation. A platform system of a collaborative waiter robot task is developed.In a preliminary experiment, two modes, namely, interactive and manual modes, of the system are compared to evaluate the effectiveness of the robot's autonomous function. The results indicate that the robot's autonomous function works better in interactive mode than in the manual mode under some conditions.

References

1. Thomaza, A.L., Breazeal, C.: Teachable robots: Understanding human teaching behavior to build more effective robot learners. Artificial Intelligence 172(6-7), 716–737 (2008)
2. Seiji, Y., Koh, K., Takanori, K.: Mutual adaptation and adaptation gap between a human and an agent. Journal of Japanese Society for Artificial Intelligence 21(6), 648–653 (2006)
3. Xu, Y., Ueda, K., Komatsu, T., Okadome, T., Hattori, T., Sumi, Y., Nishida, T.: Woz experiments for understanding mutual adaptation. Journal of AI& Society 23(2), 201–212 (2009)
4. Xu, Y., Ohmoto, Y., UedaA, K., Komatsu, T., Okadome, T., Kamei, K., Okada, S., Sumi, Y., Nishida, T.: Two-layered communicative protocol model in a cooperative directional guidance task. In: The 7th International Workshop on Social Intelligence Design, San Juan, Puerto Rico (2008)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience Publication, Hoboken (2001)
6. Leyton-Brown, K., Shoham, Y.: Essentials of Game Theory - A Concise, Multidisciplinary Introduction. Morgan & Claypool, San Francisco (2008)