An Knowledge-level Approach for Building Human-machine Cooperative Environment

H. Takeda, N. Kobayashi, Y. Matsubara, and T. Nishida

Graduate School of Information Science,
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara 630-01, Japan
takeda@is.aist-nara.ac.jp
http://ai-www.aist-nara.ac.jp/

Abstract. In this paper, we propose the *knowledgeable environment* as a framework for integrated systems for human-machine co-existing space. In the knowledgeable environment, integration is achieved as knowledge-level communication and cooperation. We abstract all machines as information agents whose roles and communication are understandable for people. Ontology is here used as explicit representation of the abstraction of the environment which includes agents, objects, and activities of agents and people. Cooperation is also achieved by using ontology. We re-define concept of human-machine interaction in the light of knowledge-level interaction, i.e., interaction with various logical and spatial relation among participants. We realized a prototype of the knowledgeable environment with two mobile robots, rack and door agents, and demonstrated how cooperation among robots, machines and people could be implemented.

Keyword: real-world agent, ontology, mediation, cooperation, human-robot interaction

1 Introduction

In recent years, various types of computers and computer-controlled machines are introduced into our daily life. It is not distant future when so-called robots be also introduced there. Such machines are expected to improve quality of our life because they can provide performance which we cannot have but want to enhance. But introduction of these machines currently makes us annoying because of variety of their behavior and interface, i.e, each of them has its behavior and interface to user and requires us to understand them. It implies that there need intelligence that should shorten distance between human and machines and integration that should flatten complicity due to variety of machines. In short, we need a framework for integrated intelligent systems for human-machine co-existing space. In this paper, we propose the *knowledgeable environment* in which integration is achieved as knowledge-level communication and cooperation.

It is a new and challenging field for robotics, artificial intelligence, and human interface domains to deal with space for human activity. One reason for it is dynamics in physical space. Distributed and therefore cooperative systems are needed to capture spatially distributed human activities. The other reason is that human activities cannot intrinsically modeled in computers. It implies that human-machine interaction is an indispensable issue that can bridge human and computer activities. We can summarize these problems as the following three research issues;

1. Modeling of environments including machines and people: Its meaning has two-holds. One is to model not only machines and environments but also people. We cannot have perfect models of human activities as we mentioned, but partial models are still important to capture human-machine co-existing space. The other is to make models of environments understandable to humans, i.e., models are not only for machines but also humans. It is natural because people are also participants of the environments for which models are provided.

2. Extension of basis of human-machine interaction: Various and distributed sensors to detect human activities and presentation methods to people are needed to realize natural human-machine interaction in human-machine co-existing environment. One approach is to extend variety of physical instruments[9]. The other approach is to extend concept of sensoring and presenting. For example, we can call tracking of movement of people[13] as a sensor. Our three distinction of human-machine interaction (described Section 6) is a proposal for this approach.

3. Cooperative architecture for real-time distributed problems: People and machines are spatially distributed and synchronized, It means that two types of spatial distribution and synchronization exist, i.e., those for environments (machines are distributed and synchronized) and those for problems (human activities are distributed and synchronized). We need cooperative systems to integrate machines and people in such situation.

2 The Knowledgeable environment

Our approach called the *knowledgeable environment* is aiming to build a framework for cooperative systems for human-machine co-existing space. Figure 1 shows an image of space which we want to realize. In the space, people and machine are mutually cooperative, i.e., people can ask some tasks to machines and vice versa. It may seem strange that machines can ask something to people. It may be possible to assume that machines are almighty in environments like factories because lack of existence of human enables environments to be designed solely for machines. Since environments like our living space cannot be designed solely for machines, some tasks cannot be achieved only by machines but by combination of machines and people. In such case, people can be asked by machines.

In the knowledgeable environment, the above three problems is solved by knowledge-level modeling and interaction. We abstract all machines as *agents* whose roles and communication are understandable for people. Ontology is here used as explicit representation of the abstraction. Cooperation is achieved by

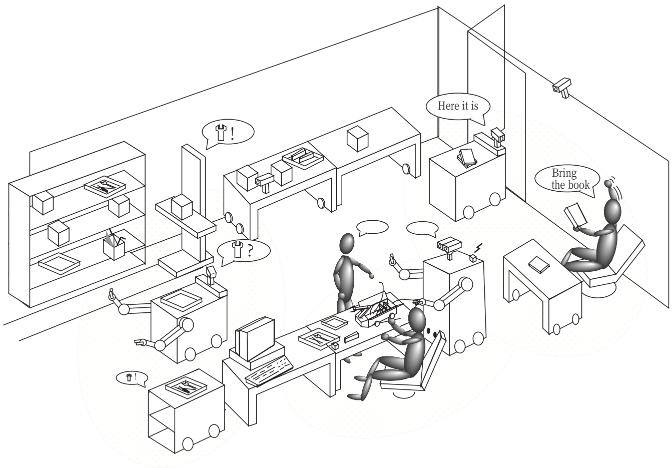

Fig. 1. An image for the knowledgeable environment

using ontology. We re-define concept of human-machine interaction in the light of knowledge-level interaction, i.e., interaction with various logical and spatial relation among participants. In this paper, we discuss the following four methods that we are currently developing in order to realize the knowledgeable environment.

1. Agentification of robots and machines
2. Ontology as modeling of environment
3. Ontology-based task mediation for cooperation
4. Various interaction between human and robots from intimate to cooperative

3 Agentification of robots and machines

The basic idea of our approach for modeling machines is to model them as software agents that can communicate to each other with some abstracted language. The merit of this approach is as follows;

- Abstracted definition of agents is applicable
- Techniques developed for software agents like cooperation are available
- Cooperation between software agents and machines is solved in a single architecture

We call agents that have facilities to obtain information from the physical environment or to do something to the environment as *real-world agents*. On

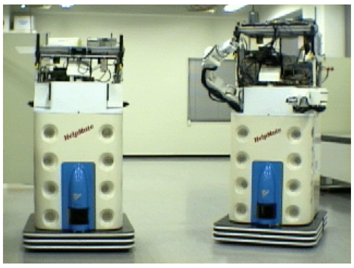

Fig. 2. Two mobile robots

Fig. 3. Rack and door agents

the other hand, we call agents concerning only information in computers as *information agents*.

All robots and machines are agentified as KQML agents[4]. KQML(Knowledge Query and Manipulation Language) is a protocol for exchanging information and knowledge among agents. KQML is mainly designed for knowledge sharing through agent communication. A KQML message consists of a message type called *performative* like *ask, tell* and *subscribe*, and a number of parameters like *sender, receiver, content* and *language*. For example, a message content is written as a value of parameter *content*. We mostly use KIF (Knowledge Interchange Format)[5] as language to describe message contents. KIF is a language for interchanging of knowledge and based on the first-order predicate logic.

A real-world agent can consist of some sub-agents each of which performs specific information processing in the agent. By separating facilities in an real-world agent, we can design agents without depending computational performance of each robot or machine. A typical real-world agent consists of three sub-agents, namely *KQML handling sub-agent* that parses and generates KQML messages, *database sub-agent* that holds status of the agent itself and its environment, and *hardware controlling sub-agent* that sends commands to actuators and obtains sensor values.

We currently agentified a mobile robot with two manipulators called *Kappa1a*, a mobile robot without manipulators called *Kappa1b* (see Figure 2), computer-controlled rack and door(see Figure 3). A manipulator has six degrees of freedom and a gripper.

We also treat human as agents to some extents. We can provide action knowledge that includes human as participants of actions. As we will describe in the next section, ontology is provided as common vocabulary between human and computer agents as well as among agents. But there are many difficulty to treat human just as computer agents. In the context of our definition of agents, difference of way of communication is crucial. Human has various communication channels that are to choose depending on situations, while computer agents have a single channel. It implies that modeling of human-machine interaction as inter-agent interaction is needed. We will discuss how human-machine interaction is modeled and integrated in the knowledgeable environment in Section 6.

4 Ontology as modeling of environment

Our aim is to establish an information infrastructure to cooperate heterogeneous real-world agents at knowledge level, i.e., to clarify what knowledge is needed for those agents for cooperation. We introduce ontologies for object, space, and action as partially shared systems of concepts among agents. These ontologies are defined for knowledge on object, action, and agents' abilities that are used in mediating given tasks (see Section 5).

4.1 Need for sharing concepts

The simplest way to accomplish a task with multiple agents is to break down the task and design subtasks each of which is executable for each agent. But this approach is not applicable where tasks are dynamically generated like environments where human and agents co-exist.

In order to do it more intelligently, agents should understand what parters are doing or requesting and so on. In other words, agents should have common communication capabilities to tell and understand intension. It means that they should share not only protocols and languages to communicate but also concepts used in their communication. The latter is called ontology which a system of concepts shared by agents to communicate to each other[6].

Ontology is defined and used mostly in information agents (For example see [10][2]). The primary concern in such studies is to model objects which agents should handle. Modeling objects is not sufficient to realize communication among real-world agents. Modeling space is also important because they should share space to cooperate each other. Modeling action is another important concept because they should understand what other agents do or request[1]. Therefore there are three ontologies, namely ontologies for object, space, and action.

4.2 Concept for object

The environments are usually fulfilled with various objects, and tasks are usually related to some of these objects. They should share concepts for objects, otherwise they cannot tell what they recognize or handle to other agents.

Difficulty lies that what they can understand are different because the way they can perceive objects is different. It depends on abilities for sensing, acting, and information processing.

The policy for making shared concepts is using abstraction levels to represent objects. We build a taxonomy of objects as hierarchy of *is-a* relations. It does not mean that all agents can understand all objects in the taxonomy. Most agents can only understand subsets of those objects because their recognition abilities are limited. For example, some agent can recognize a box but cannot recognize difference between a trash box and a parts-case, because it can only

[1] Another important concept is one for time. In this paper, time is not explicitly described but embedded as shared actions.

detect whether it is a box or not. It is sufficient for this agent to understand concept *box* and its higher concepts.

We provide current position, default position, color, weight for attributes which are common for all objects. Descriptions of attributes have also levels of abstraction.

4.3 Concept for space

The next important concept for cooperation is concept for space. Since agents are working in the physical world, they should understand space, that is, where they are, where they are moving for, where the target object exists, and so on. Especially it is important for agents to work together. According to sensing and processing abilities of agents, they can have different ways to represent space. For example, agents which move by programmed paths would represent space by paths and points to stop. Some agents can have absolute positions but some can have only relative positions. We provide the following two types of representation as shared space ontology.

1. **Representation with preposition**

   Relative position is described as combination of preposition and object which is represented as object ontology[7]. We provide seven prepositions, i.e., *at, on, in, in-front-of, behind, to-the-right-of*, and *to-the-left-of*. For example, a position in front of the rack agent is represented as `in-front-of (rack-agent)`. Actual position is determined by agents who interpret representation.
2. **Representation with action**
   A relative position can be also represented as association to actions that can be performed at. For example, to describe space as "where you can look at the rack" or "where you can meet Agent X." is useful for agents who want to achieve these actions. Actual positions may be different according to agents that would take action, because ability of action that agent can do may be different. But no matter actual positions may differ, it is sufficient to understand positions where such actions can be done.
   We describe a position with combination of an action-related term and object(s). For example, `viewpoint(rack)` means a position where the agent can look at the rack, and `meetingpoint(agent1, agent2)` means where agent1 and agent2 can meet.

4.4 Concept for action

The last category for shared concepts is concept for action. Agents should understand what other agents are doing in order to cooperate with them. Like the other categories, concepts which an agent can understand are also different according to ability of the agent itself. It is obvious that concepts which are directly associated to its physical actions. But more abstract concepts can be shared among agents. Concepts associated to agents' physical actions should be related to more abstract concepts shared by them in order to understand each other.

Definition of concept for action consists of a name, attributes like subject and starting-point, and constraints among attributes. Constraints are represented as sharing of attribute values. Relation among concepts is decomposition relation, i.e., an action can have an sequence of action which can achieve the original action. Relation among decomposed actions are represented as constraints among attributes.

5 Ontology-based task mediation for cooperation

In this section, we discuss how to realize interaction among agents with different ontologies. We introduce mediators which can break down and translate tasks to a sequence of actions each of which some agent can understand and execute.

The function of mediators here is to bridge a gap between tasks provided by human and actions that can be done by real-world agents. Since tasks should be performed cooperatively by multiple agents in most cases, tasks should be decomposed into subtasks and distributed to agents. Ontologies have two roles in this process. Firstly, it is used to understand the given tasks. Since given tasks are what humans want agents to do, they are insufficient and incomplete for specifying actions of agents. Ontologies can supply necessary information on environments and agents to complete task descriptions. Secondly, it is used to distribute tasks to agents. As we mentioned in the previous section, each agent has its own ontology which is dependent on their physical and information ability. But shared ontologies can integrate these agent ontologies using abstraction. Tasks can be translated to a set of local tasks each of which is understandable by some agent by using multiple abstraction levels in ontologies.

We realized process of the mediation by the following four steps (see Figure 4). A task is described as an incomplete description of action. Incompleteness means that all properties should not be specified, i.e, some properties are specified, but others not. Unspecified properties will be fulfilled by mediators using the current state of the environment by consulting object ontology and object knowledge, e.g., where objects are now.

Supplement of object attributes If necessary attributes of objects are missing in a task description, the mediator can add these attributes using default values in object ontology.

Assignment of agents The mediator tries to assign an agent to perform the action to realize the task. It is done by consulting knowledge on agent abilities which is represented by object, space, and action ontologies.

Action decomposition The mediator decomposes the given action into actions each of which may be executable by some agents. Decomposition of action is done by consulting action ontology. Action decomposition and agent assignment are done simultaneously because action decomposition restricts

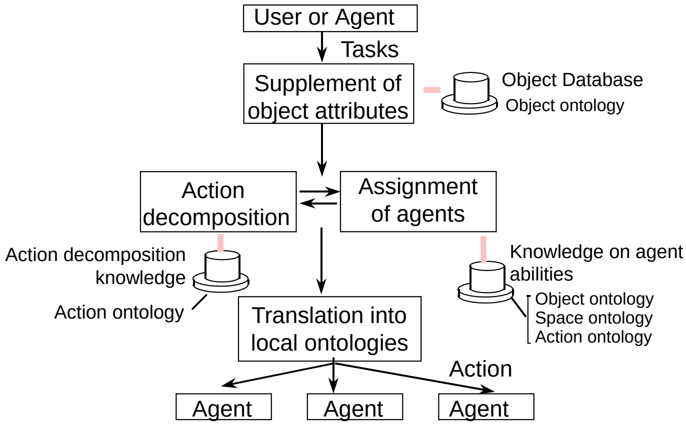

Fig. 4. Mediation flow

agent assignment and vice versa. If all actions are assigned to executable agents, the both steps are finished.

Translation into local ontology All information before this step is represented by the shared ontologies. Before sending out the decomposed actions to agents as a message, the mediator translates each message into one in the local ontology to the receiver agent.

The implemented mediator has two parts, i.e, planner and executor. Planner processes the above mediating steps, and executor binds agents participating the action sequence and control executing sequence of action (see Figure 5).

The above process describes how to deal with a single task. In the human-machine co-existing environment, there are multiple asynchronous tasks. In our approach, it is processed by cooperation among multiple mediators. The basic idea is that every emerged task invokes a mediator and then each mediator tries to gather and control necessary agents independently. Each mediator processes the given task by using state information of the environment and communication with other mediators if necessary (see Figure 5).

6 Various interaction between human and robots

We need natural ways for people to communicate and cooperate with machines or robots just as same as they do with other people, i.e., people interact with other people anywhere at anytime. In this section, we mainly focus on interaction between people and mobile robots.

The primitive way for human-robot interaction is interaction through special instruments. People can communicate with robots by using instruments like computers. Recent technologies for multimodal communication can provide various

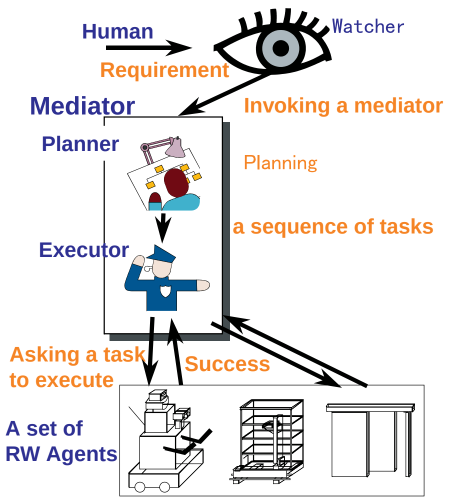

Fig. 5. Behavior of mediator(1)

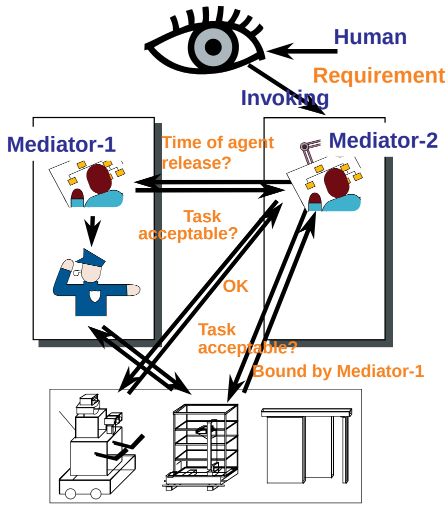

Fig. 6. Behavior of mediator(2)

communication channels like voice and gestures(e.g., [3]). Interface agents (e.g., [8])can be used for their communication. But people could not communicate with robots directly, and they are bound to computer terminals.

Other way is direct interaction with people and robots. In addition to multimodal communication with computer, robots can use their bodies when they communicate to people. Although it is more restricted in expressive flexibility than *virtual* interface agents because of their mechanical structures, physical motion are more natural and acceptable for people. We call such direct interaction between robot and people *intimate interaction*.

The intimate interaction can involve people in multimodal direct interaction, but another problem arises. People and robots should be close to each other in order to establish such interaction. It is obstacle to realize ubiquitous interaction among people and robots. We need interaction between people and robots who are separate from each other. We call such interaction *loose interaction*.

Loose interaction absorbs the distance problem between people and robots, but interaction is still closed within participants of interaction. We sometimes need more robots (or even people) involved to accomplish interaction. For example, a robot is asked to bring a book by a person, but it has no capacity to bring books. It should ask another robot that can bring books and the person should interact another robot as a result. We call this type of interaction *cooperative interaction*. Cooperative interaction makes interaction extensive, i.e., interaction can be extended by introducing more robots and people as much as it needs. It can solve the problem of limitation of functions of each robot so that interaction should not be bound to functions of robots that people are interacting.

6.1 Intimate human-robot interaction

The first type of interaction we investigate is intimate interaction which is direct one-to-one interaction between people and robots. We provide two communication channels, i.e., gesture and vocal communication. People can tell their intention by using their gestures, and the real-world agent can tell its intention by its gestures and voice.

Gesture recognition is implemented in a relatively simple way, i.e, we can extract gestures only by hands. Firstly the agent identifies motion areas of hands by searching a black part in the scene and assuming it person's head. Secondly, it defines rectangle areas adjacent to both sides of the black part as motion areas of hands. Thirdly, it detects motion of hands by optical flow. The result is sequences of primitive hand motions which are specified by hand and direction. Then gestures are identified by comparing detected sequences of motions with knowledge on gestures. We provide some gestures like "shake", "wave", and "move both hands".

There needs another step to know meaning of such detected gestures, because meaning of gestures is dependent on situation of interaction. In our system, the real-world agent reacts to gestures according to predefined state transition network. Each state has actions that the real-world agent should take and some links to other states. Each link has conditions described with gestures of the person and its sensor modes. If one of conditions of link of the current state is satisfied, the current state is shifted to next state that is pointed by the link. Since a single gesture can be included in conditions of multiple links, multiple interpretation of gestures is possible. Figure 7 shows an example of intimate interaction.

Variety of actions that real-world agents can perform are classified into two. One is informative actions or gestures which cause no physical changes of the environment like "Yes", "No", and "Ununderstanding" using head motion, and "bye-bye" and "raise both hands" using hand motion. Voice generation is also included in possible informative actions of the real-world agent. The other is effective actions which cause physical changes of the environment like "grasp something" and "release something" using hand motion, and "move to somewhere" using driving units.

We currently provide some interaction modes like "take a box", "*janken*[2]", and "bye-bye". Some interaction is closed within the real-world agent and the person, but others not. If the latter case, the real-world agent should ask tasks to a mediator in order to involve other real-world agents. We will discuss this process as cooperative interaction later.

6.2 Loose human-robot interaction

Loose interaction is interaction between people and robots who are separated. Since robot may not see the person, the same method for intimate interaction is

[2] It is a children's game in which two or more person show one of three forms of hand to each other. The agent uses hand motions instead of forming hands.

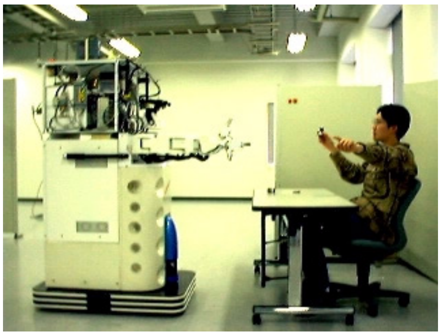

Fig. 7. An example of intimate interaction ("throw it out")

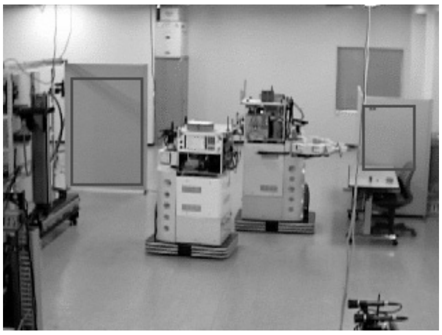

Fig. 8. Scene by camera for watcher (two boxes are where watcher is "watching".)

not applicable. We introduce an agent called "watcher" which *watches* a room to find what is happening in the room. It uses a camera to look over the room (see Figure 8) and communication to other agents.

If the watcher notices a request from someone to others, it composes a task description and passes to a mediator. Notification of requests comes by either recognition of camera scenes or communication from other agents. the watcher currently observes two areas, i.e., around a door and a desk (two boxes in Figure 8). An example of knowledge on task composition is shown in Figure 9. This definition tells "if it is found by camera that someone is waving, compose a task that Kappa1a should go to her/his position". As a result, the person who waves can tell her/his intention to the real-world agent even if it is not near her/him (see Figure 10). It is important that the watcher should not make direct order to real-world agents but tasks which can be scheduled by mediator. If the appointed agents are busy to process other tasks, the mediator can determine that the composed task may be postponed until the current task is finished, or be processed by other agents.

6.3 Cooperative human-robot interaction

Interaction should sometimes be extended to include agents needed to accomplish its purpose, i.e., interaction should be performed cooperatively by more than two agents. Suppose that a person is facing a robot that cannot take and carry objects and asking the robot to bring an object to her/him. The robot may try to do it by itself and finally finds it cannot, or simply refuse her/his request because it knows that it is impossible for it to do it. A better solution is that the robot should ask other robots that can take and carry objects to perform the recognized request. In this case, three agents, i.e., a person and two robots are necessary members to accomplish the interaction.

Cooperative human-robot interaction is realized here by mediators. If requests are detected by cameras, this process is done by watcher(see Figure 5).

```
(define Come_on
 (content
  ((behavior wave)
   (source   camera)
   (client   ?human)))
 (task
  ((subject camera)
   (come (subject kappa1a)
   (destination ?human)))))
```

Fig. 9. Knowledge on task composition

Fig. 10. An example of loose interaction (a camera behind the robot detected human request and told the robot to go)

Otherwise requesting agents themselves compose tasks and send them to the watcher. Then the watcher invokes a mediator and delegates the task to it.

Figure 11 shows how the example of cooperative interaction mentioned above can be solved in our system. In this example, two mediators are generated to solve a task with two mobile agents, a rack agent, and a person. In the example, the person asked a mobile agent to bring a manual on the rack. Unfortunately the mobile agent could not take objects on the rack. Then it asked to the watcher to solve the task. The mediator invoked by the watcher made a plan and executed it. On the other hand, watcher made another task to delegate another mediator because the mobile agent was obstacle for the first plan.

7 Related work

Most relevant studies are Robotic room[9] and Intelligent room[13][1]. Although they have similar goals but their methods are different in according to their application fields.

Robotic room is aiming intelligent environments for health care or hospitals. The key technology is to provide various sensoring devices and to detect human behaviors with them. It is different approach to ours in treatment of people in the system. People in their research are something for the system to observe, which is analogous to patients in hospitals.

Intelligent room project investigates various computational techniques to support people in meeting or discussion, for example, tracking person's movement and augmented reality that can impose computer-generated images to real images. People are here active and the system tries to help their activities, which is analogous to tools in offices.

On the other hand, the system and people are mutually understandable and cooperative in our system. Not only people can ask the system to help them,

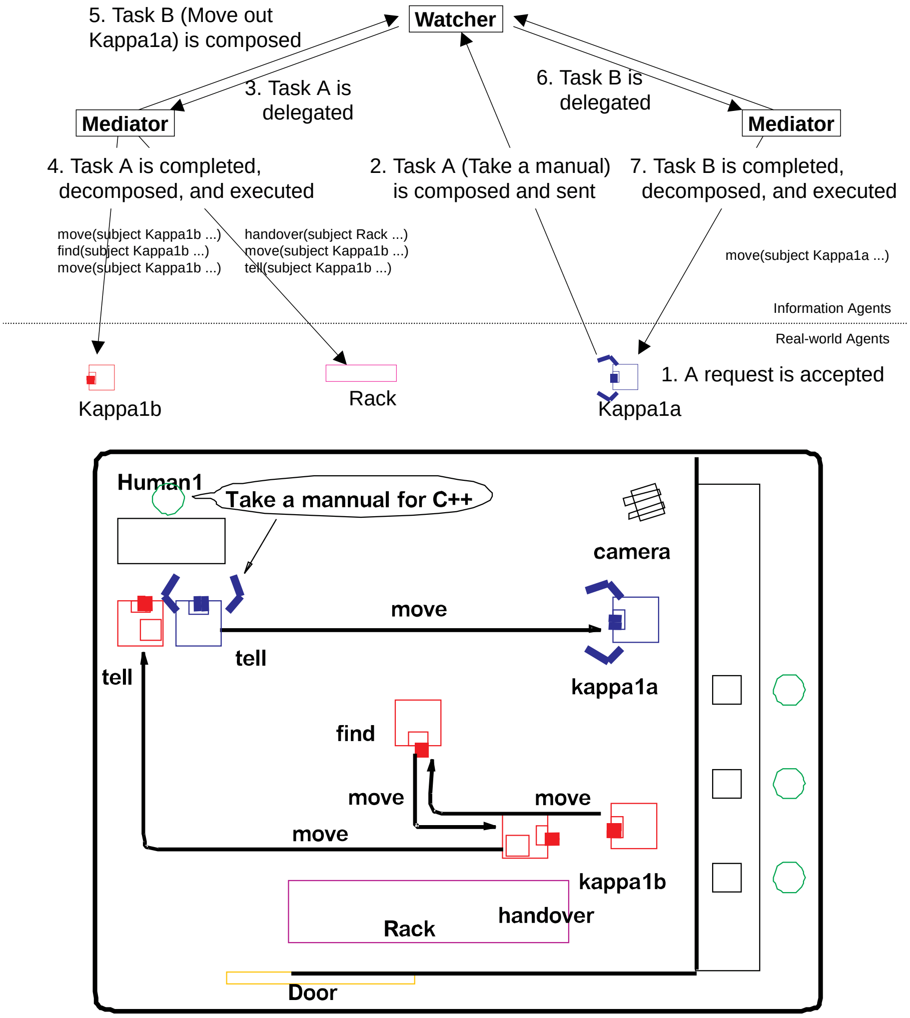

Fig. 11. An example of cooperative interaction

but the system may request people to help it when they are unable to perform the task asked by people. It is analogous to partners or secretaries in office.

It is interdisciplinary work so that there are much related work in artificial intelligence, robotics, and human interfaces. In particular there are interesting studies on human-robot interaction and cooperation of agents. Please see Takeda[12] in detail.

8 Conclusion and future work

We proposed the knowledgeable environment in which all machines and interaction between machine and people are modeled as knowledge-level communication. We provide ontology for basis of communication and mediation of tasks based on the ontology. Human-machine interaction is realized as three different ways that can be applied depending on physical and logical relation between people and machines.

One of the key issues in our approach is how to provide good ontology for human-machine co-existing space. The current ontology is naive and poor in describing various machines and human activities. We should investigate them more precisely. For example, human actions and their utilization of objects in ordinary office work is needed to analyze. Cooperation of agents is still insufficient, especially we should consider more tight cooperation in order to be applicable to situations with more limited resources.

References

1. M. H. Coen. Building brains for rooms: Designing distributed software agents. In *IAAI-97*, pages 971–977, 1997.
2. M. R. Cutkosky, R. S. Engelmore, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenenbaum, and J. C. Weber. PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, January 1993:28–38, 1993.
3. T. Darrell and A. Pentland. Space-time gestures. In *Proceedings of IEEE 1993 Computer Society Conference on Computer Vision and Pattern Recognition*, pages 335–340, 1993.
4. T. Finin, D. McKay, R. Fritzson, and R. McEntire. KQML: An information and knowledge exchange protocol. In K. Fuchi and T. Yokoi, editors, *Knowledge Building and Knowledge Sharing*. Ohmsha and IOS Press, 1994.
5. M. Genesereth and R. E. Fikes. Knowledge interchange format, version 3.0 reference manual. Technical Report Technical Report Logic-92-1, Computer Science Department, Stanford University, June 1992.
6. T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, August 1993.
7. A. Herkovits. *Language and spatial cognition.* Cambridge University Press, 1986.
8. P. Maes and R. Kozierok. Learning interface agents. In *Proceedings of AAAI-93*, pages 459–465, 1993.
9. T. Sato, Y. Nishida, J. Ichikawa, Y. Hatamura, and H. Mizoguchi. Active understanding of human intention by a robot through monitoring of human behavior. In *Proceedings of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 405–414, 1994.
10. H. Takeda, K. Iino, and T. Nishida. Agent organization and communication with multiple ontologies. *International Journal of Cooperative Information Systems*, 4(4):321–337, December 1995.

11. H. Takeda, K. Iwata, M. Takaai, A. Sawada, and T. Nishida. An ontology-based cooperative environment for real-world agents. In *Proceedings of Second International Conference on Multiagent Systems*, pages 353–360, 1996.
12. H. Takeda, N. Kobayashi, Y. Matsubara, and T. Nishida. Towards ubiquitous human-robot interaction. In *Working Notes for IJCAI-97 Workshop on Intelligent Multimodal Systems*, pages 1–8, 1997.
13. M. C. Torrance. Advances in human-computer interaction: The intelligent room. In *Working Notes of the CHI 95 Research Symposium*, Denvar, Colorado, 1995.