The Knowledgeable Environment

— An Knowledge-level Approach for Human-machine Co-existing Space —

Hideaki Takeda, Nobuhide Kobayashi, Yoshiyuki Matsubara, and Toyoaki 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.

Introduction

In recent years, various types of computers and computer-controlled machines are introduced into our daily life. We expect so-called robots will be also introduced there soon. Such machines contribute to improve quality of our life because they provide performance which we cannot have or want to enhance. But introduction of these machines also make us annoying because each of them has its behavior and interface to user and requires us to understand them. We need a framework for integrated 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.

Requirements for systems for human-machine co-existing space

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

1. Modeling of environments which include machines and people
   Its meaning has two-holds. One is to model not only machines but also people. As we mentioned above, we cannot have perfect models of human activities but can have models of human activities to some extent. The other is to make models of environments which are understandable 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(Sato *et al.* 1994). The other approach is to extend concept of sensoring and presenting. For example, we can call tracking of movement of people(Torrance 1995) as a sensor. Our three distinction of human-machine interaction (described Section 7) 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.

An knowledge-level approach for human-machine co-existing space

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

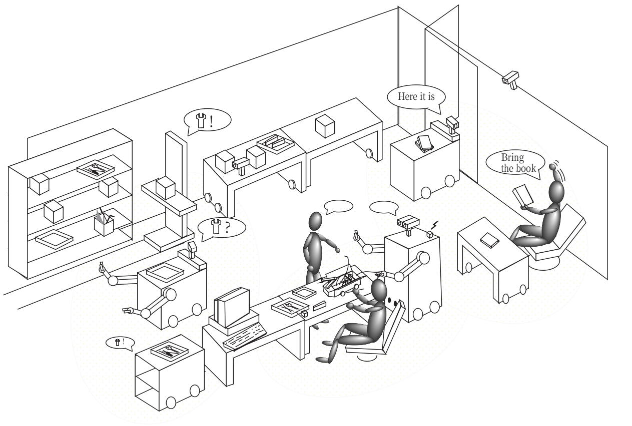

Figure 1: An image for the knowledgeable environment

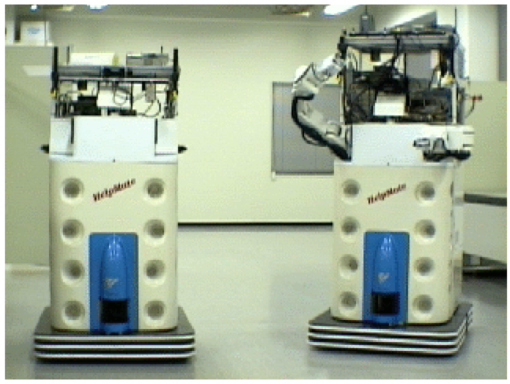

Figure 2: Two mobile robots

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. Since machines in our daily living space cannot be almighty unlike those in factories, 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 information agents whose roles and communication are understandable for people. Ontology is here used as explicit representation of the abstraction. Cooperation is 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.

In the following section, we show the current progress of our research(see (Takeda *et al.* 1996) and (Takeda *et al.* 1997)).

Multi-agent architecture for real-world agents

The basic idea of our approach for modeling machines is to model them as software agents, which 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 the same architecture

We call agents which have facilities to obtain information from the physical environment or to do something to the environment as *real-world agents*. On the other hand, we call agents concerning only information in computers as information agents.

All robots and machines are agentified as KQML agents(Finin *et al.* 1994). 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)(Genesereth & Fikes 1992) as language to describe message contents. KIF is a language for interchange of knowledge and based on the first-order predicate logic.

A real-world agent can consist of some sub-agents each of which perform specific information processing in the agent. By separating facilities in an agent, we can construct agents without depending physical performance of each robot or machine. A typical real-world agent consists of three sub-agents, namely *KQML handling sub-agent* which parses and generate KQML messages, *database sub-agent* which holds status of agent itself and environments, and *hardware controlling sub-agent* which sends commands to actuators and to obtain sensor values.

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

Knowledge for cooperation of agents

Our aim is to establish information infrastructure to cooperate heterogeneous real-world agents at knowledge level, i.e., to clarify what knowledge is needed for those agents for cooperation.

Figure 3: Rack and door agents

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 defined like environments where human and agents coexist.

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 abilities 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(Gruber 1993).

Ontology is defined and used mostly in information agents (For example see (Takeda, Iino, & Nishida 1995)(Cutkosky *et al.* 1993)). The primary concern in such studies was 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 (see Figure 4).

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 taxonomy of objects as hierarchy of *is-a* relations. It does not mean that all agents can understand all objects in this figure. Most agents can only understand subsets of those objects because 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 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. For example, in most abstract description of weight, there are only three values, i.e., *light, middle*, and *heavy*. In other abstract level, it can be represented by more symbols, or numerically. The level of abstraction is determined by abilities of sensing and pattern recognition.

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 cannot have absolute positions but relative position.

We provide the following two types of representation as shared space ontology.

Representation with preposition Relative position is described as combination of preposition and object which is represented as object ontology(Herkovits 1986). 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.

Representation with action Relative position can be also represented by action. For example, it is useful for agents who want to achieve action to describe space as "where you can look at the rack" or "where you can meet Agent X." The actual position may be different according to which agent would take action, because ability of action agent can do may be different. No matter actual position differs, it is sufficient to understand positions where such action can be done.

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

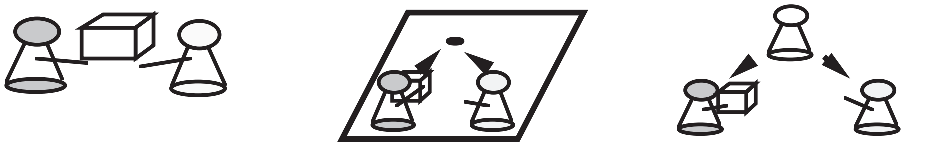

Figure 4: Three types of concepts

We describe 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.

Concept for action

The last category for shared concepts is concept for action. Agents should understand what the 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, in particular concepts which are directly associated to its physical action. 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 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.

Mediation for Task Execution

In this section, we discuss how to realize communication among agents with different ontologies. We introduce mediators which break down and translate tasks to be able to understand agents(Takeda, Iino, & Nishida 1995).

The function of mediators is to bridge a gap between tasks specified by human and actions which can be done by agents. Since in most cases, tasks should be performed by multiple agents, tasks are decomposed into subtasks to distribute 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 a sequence of actions. Ontologies supply 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 to their physical and information ability. But shared ontologies 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 provide four processes to process the given tasks in mediation (see Figure 5). A task is described as an incomplete action which has some properties like subject and object. Incompleteness means that all properties should not be specified. Unspecified properties will be fulfilled by mediators using the current state of the environment like what each agent is doing and where objects are.

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 assigns agents to perform actions 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 can 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 agent assignment and vice versa.

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

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 it 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.

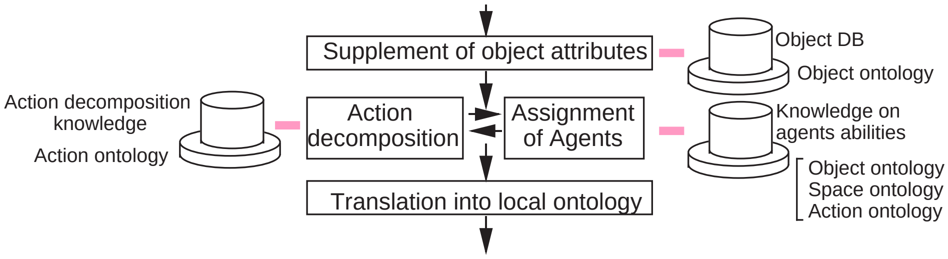

Figure 5: Mediation flow

Human-machine interaction

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 communication channels like voice and gestures(e.g., (Darrell & Pentland 1993)). Interface agents (e.g., (Maes & Kozierok 1993))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 than *virtual* interface agents because of their mechanical structures, physical motion are more natural and acceptable for people. We call such physical direct interaction between robot and people *intimate interaction*.

The intimate interaction enables people multimodal direct interaction, but another problem arises. People and robots should be close to each other to establish such interaction. It is obstacle to realize ubiquitous interaction among people and robots. We also need *loose interaction* such as interaction among people and robots who are apart from each other or interaction among people and anonymous robots which are ready to response.

Although loose interaction absorbs the distance problem between people and robots, 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 which 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 which people are interacting.

Intimate human-robot interaction

The first 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 and can only extract gestures 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 which is pointed by the link. Since a single gesture can be included in conditions of multiple links, multiple interpretation of gestures is possible. Figure 6 shows an example of intimate interaction.

The real-world agent has variety of actions. 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

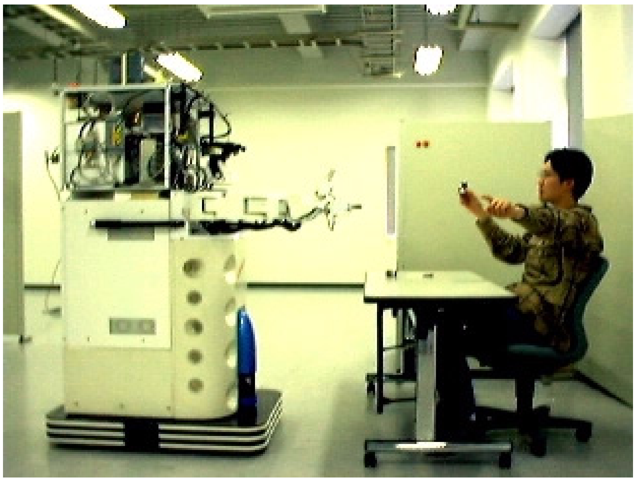

Figure 6: An example of intimate interaction ("throw it out")

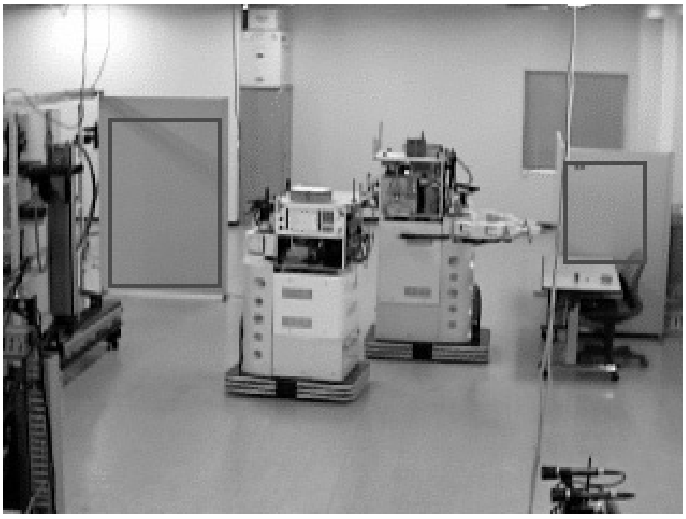

Figure 7: Scene by camera for watcher

"raise both hands" using hand motion. Voice generation is also included in possible informative actions of the real-world agent. 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 asks tasks to mediator in order to involve other real-world agents. We will discuss this process as cooperative interaction later.

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 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 7) and communication to other agents.

If watcher notices a request from someone to others, it composes a task description and passes to mediator. Notification of requests comes by either recognition of camera scenes or communication from other agents. Watcher currently observes two areas, i.e., around a door and a desk (two boxes in Figure 7). An example of knowledge on task composition is shown in Figure 8. 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 9). It is important that watcher should not make direct orders to real-world agents but tasks which can be scheduled by mediator. If the appointed agents are busy to process other tasks, the composed task may be postponed until the current task is finished, or be processed by other agents.

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

Figure 8: Knowledge on task composition

Cooperative human-robot interaction

Some interaction should 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 which 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 find 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 which can take and carry ob-

[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.

Figure 9: An example of loose interaction (a camera behind the robot detected human request and told the robot to go)

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

Figure 10 shows how the example of cooperative interaction mentioned above can be solved in our system. The following numbers correspond those in Figure 10.

1. The person asks the mobile agent (Kappa1a) in front of her/him to bring a manual for C++ by gestures.
2. The real-world agent understands the request, but finds it is impossible for it to bring the manual, because the manual is now on the rack and it cannot take objects on the rack. Then it decides to compose a task for the request and sends it to the watcher.
3. The watcher invokes a mediator newly for the task, and delegates the task to the mediator.
4. The mediator completes and decomposes the task into a sequence of actions each of which is executable for an agent. The sequence means that another mobile agent (Kappa1b) will find the position of the book, receive it from the rack agent by working together, and bring it to the person. Then it executes this sequence one by one.
5. In the above process, the mediator finds that Kappa1a would be obstacle for Kappa1b to approach the person. It composes a new task that Kappa1a would go out, and sends it to the watcher.
6. Receiving the task, the watcher invokes another mediator and delegates the task to it.
7. The secondly invoked mediator also completes and decomposes the delegated task to a sequence of actions. Then it executes them one by one.

Related work

Most relevant work are Robotic room(Sato *et al.* 1994) and Intelligent room(Torrance 1995)(Coen 1997). 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 are something for the system to observe in their system, 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 which 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, 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(Takeda *et al.* 1997) in detail.

Summary

We proposed here 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 which differ in 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.

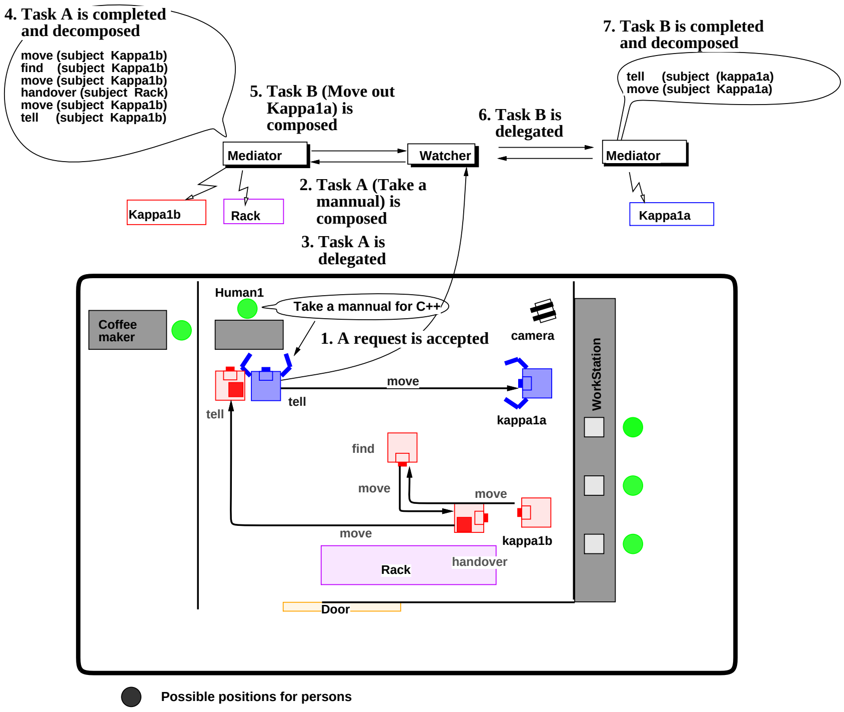

Figure 10: An example of cooperative interaction

References

Coen, M. H. 1997. Building brains for rooms: Designing distributed software agents. In *IAAI-97*, 971–977.

Cutkosky, M. R.; Engelmore, R. S.; Fikes, R. E.; Genesereth, M. R.; Gruber, T. R.; Mark, W. S.; Tenenbaum, J. M.; and Weber, J. C. 1993. PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer* January 1993:28–38.

Darrell, T., and Pentland, A. 1993. Space-time gestures. In *Proceedings of IEEE 1993 Computer Society Conference on Computer Vision and Pattern Recognition*, 335–340.

Finin, T.; McKay, D.; Fritzson, R.; and McEntire, R. 1994. KQML: An information and knowledge exchange protocol. In Fuchi, K., and Yokoi, T., eds., *Knowledge Building and Knowledge Sharing*. Ohmsha and IOS Press.

Genesereth, M., and Fikes, R. E. 1992. Knowledge interchange format, version 3.0 reference manual. Technical Report Technical Report Logic-92-1, Computer Science Department, Stanford University.

Gruber, T. R. 1993. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University.

Herkovits, A. 1986. *Language and spatial cognition*. Cambridge University Press.

Maes, P., and Kozierok, R. 1993. Learning interface agents. In *Proceedings of AAAI-93*, 459–465.

Newell, A. 1982. The knowledge level. *Artificial Intelligence* 18:82–127.

Sato, T.; Nishida, Y.; Ichikawa, J.; Hatamura, Y.; and Mizoguchi, H. 1994. 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, 405–414.

Takeda, H.; Iwata, K.; Takaai, M.; Sawada, A.; and Nishida, T. 1996. An ontology-based cooperative environment for real-world agents. In *Proceedings of Second International Conference on Multiagent Systems*, 353–360.

Takeda, H.; Kobayashi, N.; Matsubara, Y.; and Nishida, T. 1997. Towards ubiquitous human-robot interaction. In *Working Notes for IJCAI-97 Workshop on Intelligent Multimodal Systems*, 1–8.

Takeda, H.; Iino, K.; and Nishida, T. 1995. Agent organization and communication with multiple ontologies. *International Journal of Cooperative Information Systems* 4(4):321–337.

Torrance, M. C. 1995. Advances in human-computer interaction: The intelligent room. In *Working Notes of the CHI 95 Research Symposium*.