Towards Ubiquitous Human-Robot Interaction

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

Multimodality for interaction tends to be considered as use of different physical communication channels for face-to-face interaction. But our usual communication is indeed more flexible, e.g., communication to others at a distance, communication to someone who can reply, and communication with a group of people together. We categorize intimate, loose, and cooperative interaction as extended multimodal interaction. In this paper we show how such different types of interaction is realized as an integrated system with robots, people, and computers. Firstly, we introduce our multi-agent architecture to model the environment which includes people, robots, automated instruments, and computers. Each of them is modeled as agent, and all communication is realized in an inter-agent communication language. Secondly, we describe how intimate interaction is realized. We use gesture recognition, gesture generation, and speech generation for interaction. Thirdly, we introduce a software agent called "watcher" to realize loose interaction. Watcher always looks at the environment to detect whether someone is requesting interaction. It also uses gesture recognition in a coarse level. Fourthly we provide mediators for cooperative interaction. A mediator is invoked each time interaction is requested in the environment. It can gather and enroll necessary agents for interaction by planning and sometimes by consulting other mediators.

Introduction

Recent advances in computer science and robotics make robots more applicable to our daily life. Robots and people will co-exist with sharing and cooperating tasks in various ways (e.g., (Nauba, Powers, & Birchfield 1995)(Asoh *et al.* 1996)). The arising problem is how to communicate and cooperate with people. We need natural ways for people to communicate and cooperate with robots just as same as they do with other people, i.e., people interact with other people anywhere at anytime. We call such kind of communication and cooperation "ubiquitous human-robot interaction".

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.

In this paper, we show how we can realize such three different types of interaction together. In Section 2,

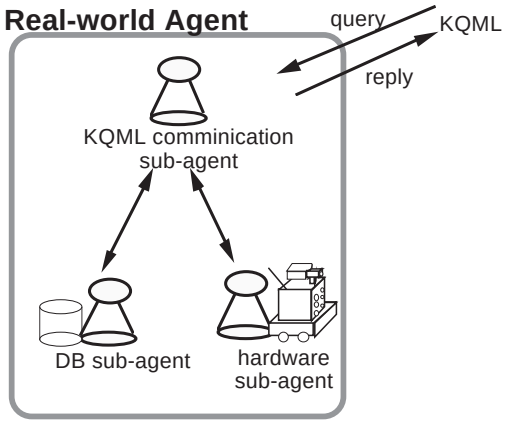

Figure 1: Architecture of real-world agents

we introduce our multi-agent architecture to model the environment which includes people, robots, automated instruments, and computers. Each of them is modeled as agent, and all communication is realized in an inter-agent communication language. In Section 3, we describe how intimate interaction is realized. We use gesture recognition, gesture generation, and speech generation for interaction which are dependent to situation of interaction. In Section 4, we introduce an agent called "watcher" to realize loose interaction. Watcher always looks at the environment to detect whether someone is requesting interaction. It also uses gesture recognition in a coarse level. In Section 5, we provide mediators for cooperative interaction. A mediator is invoked each time a task is thrown in the environment. It can gather and enroll necessary agents for the task by planning and sometimes by consulting other mediators. We discuss our approach with related work and conclude the paper in the last two sections.

Multi-agent architecture for real-world agents

We have developed multi-agent architecture for real-world agents (Takeda *et al.* 1996). The basic idea is that all participants in the environment are modeled as agents which can communicate to each other in the knowledge level. We provide shared ontologies for object, place, and action, and all agents can use them to some extent. The degree how much ontologies are understandable is dependent on physical and computational abilities of agents. Communication is realized as KQML message(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.

We provide mainly two types of agents, software agents and real-world agents. Software agents are those which do not posses physical interaction methods like cameras and motors, while real-world agent is those which are *agentified* robots and instruments by adding software modules for message interpretation and knowledge- and data-bases. Figure 1 shows a typical architecture of a real-world agent which consists of three sub-agents. KQML communication sub-agent can interpret and translate KQML messages from other agents into local messages, and vice versa. Database sub-agent can memorize states of the real-world agent, and hardware sub-agent can operate actuators like motor and obtain sensed data from sensors.

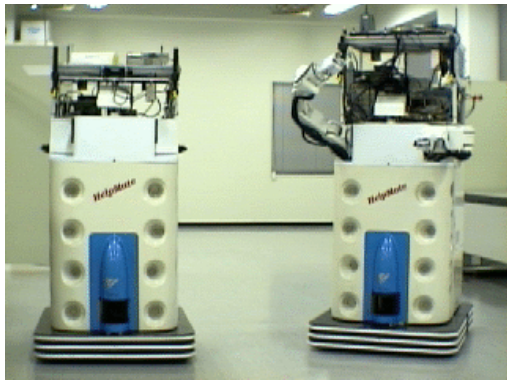

Figure 2: Two mobile robots

Figure 3: Rack and door agents

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).

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

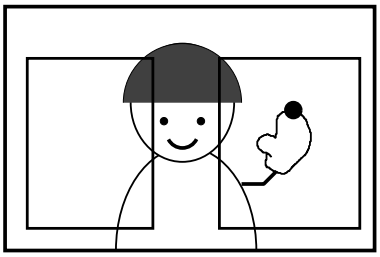

Figure 4: Extraction of motion area

```
(move (object righthand)(direct up))
(move (object lefthand)(direct down))
(move (object bothhand)(direct left))
```

Figure 5: Examples of primitive hand motions

the real-world agent can tell its intention by its gestures and voice [1].

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(see Figure 4). Thirdly, it detects motion of hands by optical flow. The result is sequences of primitive hand motions which are specified by hand and direction (see Figure 5). Then gestures are identified by comparing detected sequences of motions with knowledge on gestures. Figure 6 shows examples of 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 7 shows an example of intimate interaction.

The real-world agent has variety of actions. One is

[1] Speech recognition is not implemented yet, so people cannot tell their messages by voice. The other research group has implemented direction-sensitive speech recognition on the same robot, but it is not integrated to our system.

```
(define wave"
 (struct
  (sequence
   (
    (move (object righthand)(direct right))
    (move (object righthand)(direct left))
    (move (object righthand)(direct right))
    (move (object righthand)(direct left))
   ))))
(define shake
 (struct
  (sequence
   (
    (move (object righthand)(direct up))
    (move (object righthand)(direct down))
    (move (object righthand)(direct up))
    (move (object righthand)(direct down))
   ))))
```

Figure 6: Examples of definition of gestures

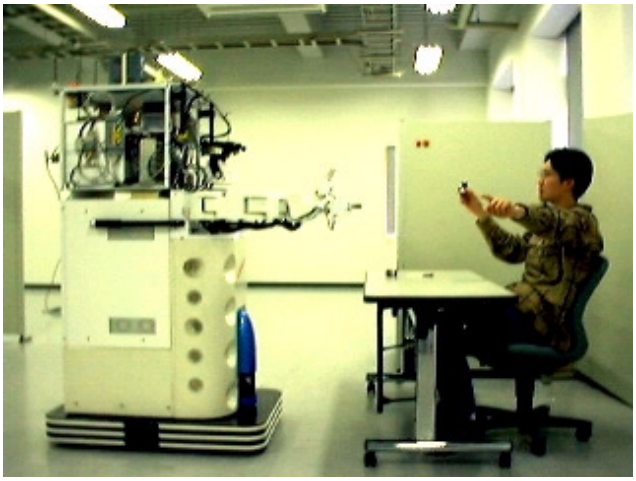

Figure 7: An example of intimate interaction ("throw it out")

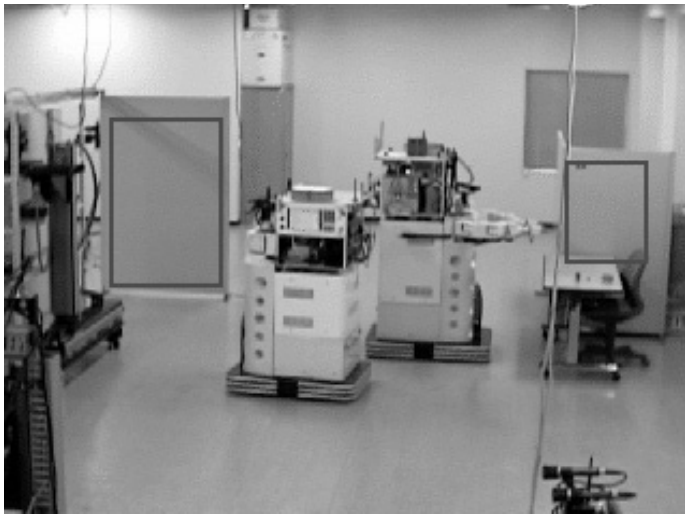

Figure 8: Scene by camera for watcher

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

Figure 9: Knowledge on task composition

informative actions or gestures which cause no physical changes of the environment like "Yes", "No", and "Ununderstanding" using head motion, and "bye-bye", "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 in Section 5.

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 8) 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. Recognition of camera scenes is achieved by the same way in gesture recognition in intimate interaction. 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 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.

[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.

Cooperative human-robot interaction

As mentioned in Section 1, 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 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.

We have developed cooperation among real-world agents using mediator (Takeda *et al.* 1996). We here extend it to process multiple asynchronous requests. The basic idea is that every emerged interaction tries to gather and control necessary agents independently. Each mediator processes a single interaction by using state information of the environment and communication with other mediators if necessary.

In cooperative interaction, we abstract requests to interaction as tasks. A task is described as an incomplete action which has some properties like subject

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

```
(fetch (object solaris)
       (destination human1))
                 ↓
((move  (subject Kappa1b)
        (from    (at  Kappa1b))
        (to      (in_front_of Rack)))
 (handover (object      solaris)
           (from_place (on Rack))
           (to_place   (on Kappa1b))
           (subject    Rack))
 (move  (subject Kappa1b)
        (from    (at           Kappa1b))
        (to      (in_front_of Human2)))
 (tell  (subject Kappa1b)
        (at      Kappa1b)
        (content (talk Here_you_are))))
```

Figure 11: An example of completion and decomposition of tasks

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. Interaction is thus formed dynamically according to the current state of environment.

Requests to interaction are processed in the following way;

1. To compose tasks to realize the given requests. 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.
2. To complete and decompose the given tasks by using knowledge on object, place, and action, and current state of the environment. This process is done by mediator, especially *planner* which is a component of mediator. The result of this process is a sequence of actions for real-world agents. Figure 11 shows an example of completion and decomposition of tasks. If some necessary agents are occupied by other mediator, the mediator asks to the occupying mediator how long it would occupy those agents. According to the answer, the mediator decides either it would wait for release of those agents, or abandon the current plan. At the latter case, it tries to make an alternative plan, otherwise it replies to the watcher failure of planning.
3. To execute sequences of actions. It is done by *executor* which is the other component of mediator. It repeats to ask agents to perform a single action and wait for its finish of all actions in the sequence. Each agent of participants of the plan is occupied by the executer from the beginning of the plan until all actions for it are finished. If any action is failed by some reason, the executor notifies failure of the plan to planner, and planner again generates a plan for the task.

Figure 12 shows how the example of cooperative interaction mentioned above can be solved in our system. The following numbers correspond those in Figure 12.

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 decompose 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 decompose the delegated task to a sequence of actions. Then it executes them one by one.

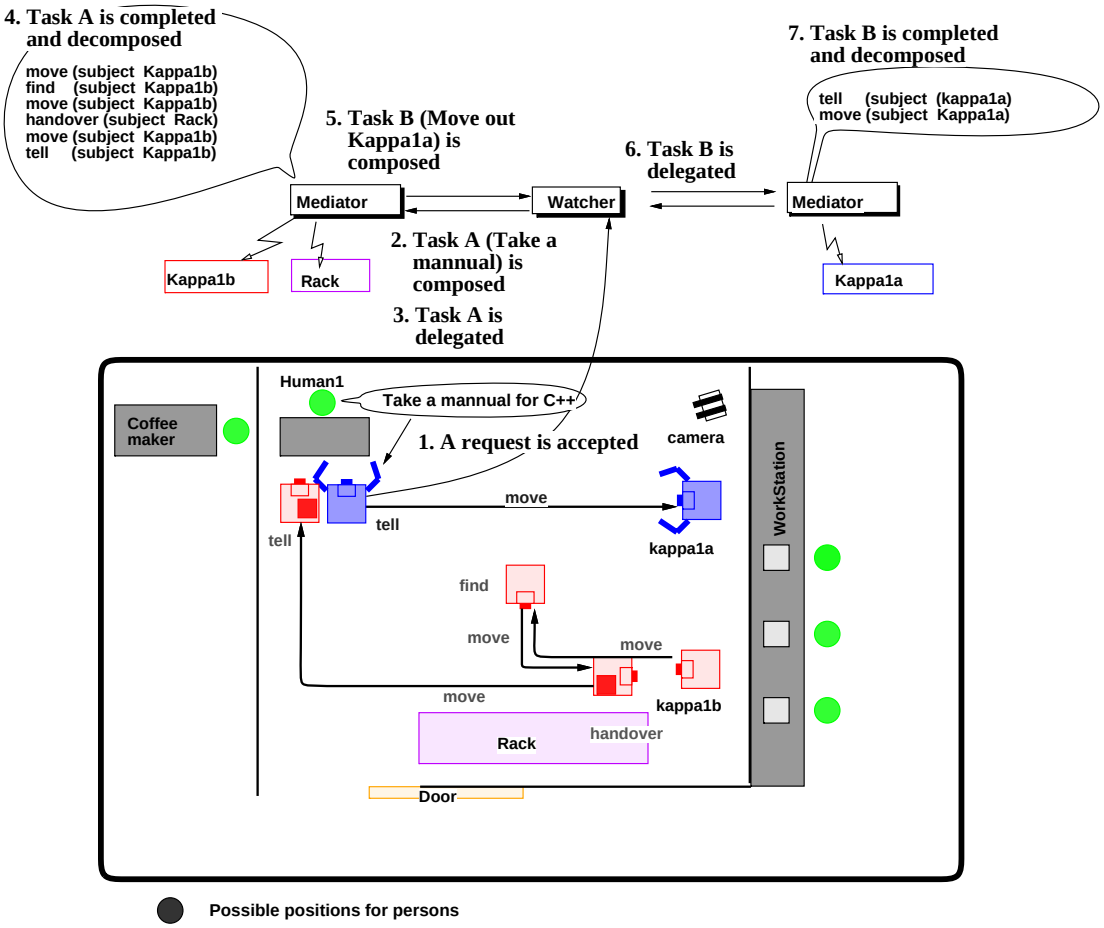

Figure 12: An example of cooperative interaction

Related work

There are many studies on robots co-existing with people. Most of them are to build autonomous robots intelligent enough to co-exist with people. For example, office conversant robot (Asoh *et al.* 1996) is aiming an autonomous robot highly integrated with learning facilities. But real environments are still too complex to learn. Our system compensates such complexity by agent communication among such robots, agentified instruments and computers.

Human-robot interaction is also an important issue for this paper. Many researchers are interested in direct human-robot interaction, especially understanding of human behavior (e.g., (Kortenkamp, Huber, & Bonasso 1996)). The other way is to capture human intention with various kinds of instruments. Sato et al.(Sato *et al.* 1994) proposed integrated use of various monitoring methods to understand human intention. Expression to people is equally important to understanding of human behavior. There are many studies on software agents for man-machine interaction from interface agent(e.g., (Maes & Kozierok 1993)) to human-like agents (e.g., (Maes 1995) and (Tosa 1995)), but physical interaction can make interaction more reliable for people (e.g., (Nakata *et al.* 1995)). As indirect human-robot interaction, Suzuki et al.(Suzuki *et al.* 1995) evaluated various kind of communication between a human operator and multi-agent robot system, but people are supervisors of robots not equal partners.

Coordination of agents is the other aspect of this paper(Cao *et al.* 1995). As centralized approach, Haigh and Veloso(Haigh & Veloso 1996) showed robot control for asynchronous and incomplete requests. As distributed approach, many methods like contract net(Smith 1980) and negotiation (Brafman & Shoham 1995) are used. Our method is medium-grain distribution, i.e., distribution by task. It is more flexible in hetrogeneousity for agents and tasks than fully distributed one because it is processed just as centralized coordination after reserving agents.

Discussion and conclusion

We showed in this paper that *multimodality* for interaction should be re-defined as that with more flexibility for participants and relations among them. Multimodality for interaction tends to be considered as use of different physical communication channels for face-to-face interaction. But our usual communication is indeed more flexible, e.g., communication to others at a distance, communication to someone who can reply, and communication with some people together. They are also our communication channels. Definition of modality for communication should include at least physical communication channels, physical relations among participants like distance, and non-physical relations among them like specified or unspecified participants and direct or indirect interaction.

Our system is an attempt for interaction with the extended modality. We model all equipments in the environment as a kind of agent, so-called real-world agent from robot to camera. Real-world agents are members of the multi-agent systems in the computer network as well as those of the physical environment. They can exchange information to each other directly, or via special software agents like mediator and watcher. The agent communication compensates for complexity of interaction in the physical environment. The physical interactions in voice and gestures is captured by the most convenient agents in the situation, and other interaction is performed by agent communication. As the result, a person in the environment should not consider which she/he should communicate to, but just communicate to something in the environment. Then the system would coordinate necessary interaction. Since intimate, loose, and cooperative interaction is thus integrated, we can say that people only have one communication channel, that is, communication to the environment.

References

Asoh, H.; Motomura, Y.; Hara, I.; Akaho, S.; Hayamizu, S.; and Matsui, T. 1996. Comibing probabilistic map and dialog for robust life-long office navigation. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, 807–812.

Brafman, R. I., and Shoham, Y. 1995. Knowledge considerations in robotics and distribution of robotics task. In *Proceedings of IJCAI-95*, 96–102.

Cao, Y. U.; Fukunaga, A. S.; Kahng, A. B.; and Meng, F. 1995. Cooperative mobile robotics: Antecedents and directions. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, 226–234.

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.

Haigh, K. Z., and Veloso, M. M. 1996. Interleaving planning and robot execution for asynchronous user requests. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, 148–155.

Kortenkamp, D.; Huber, E.; and Bonasso, R. P. 1996. Recognizing and interpreting gestures on a mobile robot. In *Proceedings of AAAI-96*, 915–921.

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

Maes, P. 1995. Artificial life meets entertainment : lifelike autonomous agents. *Communications of the ACM* 38(11).

Nakata, T.; Sato, T.; Mizoguchi, H.; and Mori, T. 1995. Synthesis of robot-to-human expressive behavior for human-robot symbiosis. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, 1608–1613.

Nauba, I.; Powers, R.; and Birchfield, S. 1995. Dervish an office-navigating robot. *AI Magazine* 16(2):53–60.

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.

Smith, R. G. 1980. The contract net protocol: Hight-level communication and control in a distributed problem solver. *IEEE transaction on Computer* C-29(12).

Suzuki, T.; Yokota, K.; Asama, H.; Kaetsu, H.; and Endo, I. 1995. Cooperation between the human operator and the multi-agent robotic system: evaluation of agent monitoring methods for the human interface system. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, 206–211.

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.

Tosa, N. 1995. Network neuro-baby with robotics hand. In Anzai., ed., *Symbiosis of Human and Artifact*, 77–82. Elsevier Science B.V.