Towards a Unified Framework for Human-Humanoid Interaction

K. Kawamura, A. Alford, K. Hambuchen, M. Wilkes, and R. A. Peters, II

Center for Intelligent Systems,
Vanderbilt University, Nashville TN 37235, USA

Abstract. In order for a humanoid robot to be accepted in society and perform as an intelligent human assistant or companion, it must be equipped with robust human-humanoid interaction (HHI) and task learning mechanisms. This paper describes our effort to achieve a robust HHI using an agent-based architecture called the Intelligent Machine Architecture (IMA). Two compound agents called the Self and the Human Agents facilitate HHI. They are aided by short- and long-term memory data structures called the Sensory EgoSphere (SES) and the Database Associative Memory (DBAM). Two applications are presented to illustrate key concepts.

1 Introduction

At the Intelligent Robotics Laboratory of the Center for Intelligent Systems at Vanderbilt University we have been developing a humanoid system called ISAC (Intelligent Soft-Arm Control) over the past several years. ISAC was originally developed for a robotic aid system for the physically disabled [10] (Figure 1). Since then it has evolved into a testbed for research in human-humanoid interaction [12, 22]. This paper describes the physical aspects of our humanoid system, the foundations for a parallel, distributed robot control architecture called the Intelligent Machine Architecture (IMA), and a framework for human-humanoid interaction (HHI). Two IMA compound agents called the Self Agent and the Human Agent facilitate HHI. Additionally, we are developing two database modules, the Sensory EgoSphere (SES) and the DataBase Associative Memory (DBAM); these modules will provide ISAC with a long-term and short-term memory, respectively.

1.1 ISAC Humanoid System

ISAC has two 6-degree-of-freedom arms actuated by McKibben artificial muscles (Figure 2). These muscles are pneumatic actuators whose lengths shorten as their internal air pressure increases [10]. They are attached to the arm in antagonistic pairs, approximating the action of human muscles. They have a significantly larger strength to weight ratio than electro-mechanical actuators; moreover, they are naturally compliant and are safe for use in close contact with people. ISAC is equipped with cost-effective anthropomorphic end effectors, built in-house. Small pistons pneumatically actuate one hand, while the

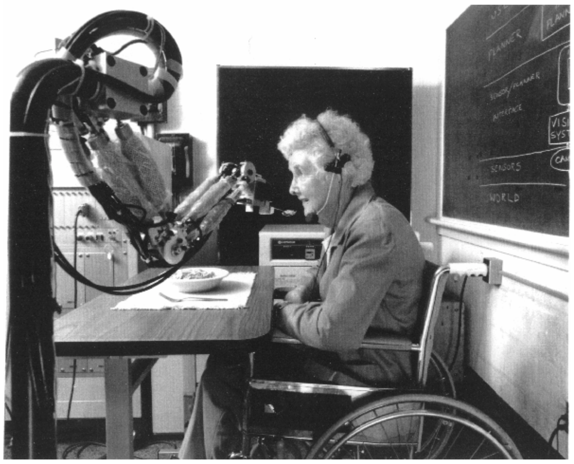

Fig. 1. Original ISAC Feeding the Physically Challenged

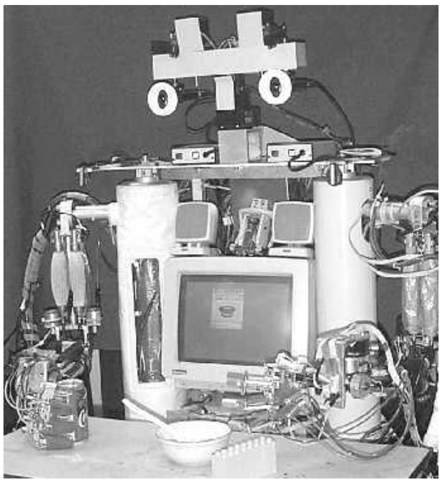

Fig. 2. ISAC Humanoid System

other employs hybrid electric/pneumatic actuation. The motors enable fine control in grasping and the piston provides strength in the grasp. The arm-hand systems have 6-axis force-torque sensors at each wrist, proximity sensors on the palms, and rudimentary touch sensors on each finger. ISAC also employs color, stereo, active vision with pan, tilt, and verge, sonic localization, and speech I/O (see http://shogun.vuse.vanderbilt.edu/CIS/IRL for more information).

2 The Intelligent Machine Architecture (IMA)

Although there is no universal agreement on the definition of an agent, one definition is a software module which seeks to achieve one or more goals. Furthermore, it is autonomous (usually meaning it controls its own actions), acts on information from its environment, and makes decisions [9].

The Intelligent Machine Architecture (IMA) is an agent-based robot control software architecture developed in-house [15]. Although the agents in IMA are autonomous, they are not necessarily intelligent. We therefore use the term *atomic agent* to describe the fundamental building blocks of the system, and to distinguish IMA agents from the more common perception of an agent as an autonomous, intelligent entity. IMA atomic agents are similar to the agents described by Minsky [14] or to the simple agents that some authors call actors [9, 18].

Atomic agents in IMA encapsulate a single aspect of the system: a piece of hardware, an object in the environment, a robot behavior, or a task. The atomic agent tightly encapsulates all aspects of that element, much as in object-oriented systems. The atomic agents execute asynchronously on separate machines. IMA supports inter-agent communication by using the Distributed Component Object Model (DCOM) [19].

A brief description of the various building blocks of IMA is given below:

Components: These are DCOM objects from which atomic agents are built. These encapsulate algorithms, data types, and communication channels.

Atomic Agents: These are composed of components. They have one or more threads of execution and are independent, autonomous entities. Usually, an atomic agent is not able to perform useful activity by itself; thus, collections of agents that communicate and interact are used to achieve useful results. There are four basic types of atomic agents, plus there is a fifth type of atomic agent that exists as a concession to realistic implementation considerations.

1. Hardware/Resource Agent: Interface to sensor or actuator hardware
2. Behavior/Skill Agent: Encapsulate basic behaviors or skills
3. Environment Agent: Provide an abstraction for dealing with objects in the robot's environment
4. Sequencer Agent: Perform a sequence of operations
5. Multi-Type Agent: Combine the functionality of at least two of the first four agent types. For example, in the interests of efficiency of implementation it may be advantageous to combine the hardware and behavior types into a single multi-type agent.

Compound Agents: Tightly-coupled group of atomic agents that are coordinated or sequenced by one or more sequencer agents.

3 Framework for Human-Humanoid Interaction for ISAC

Our philosophy for humanoid software design is to integrate both humans and the humanoid in a unified framework [11]. For any given humanoid system, the development of human-humanoid interation (HHI) will depend on certain aspects of both the human and the humanoid. In Table 1, we list the aspects we considered in designing HHI for ISAC.

ASPECTS	*Human*	*Humanoid*
Physical	Face	Structure
	Hand	Workspace
Sensory	Audio	Audio
	Vision	Vision
	Others	Tactile
		Infrared
Cognitive	Emotion	Affect
	Logic	DBAM
		SES

Table 1. Aspects of a Human-Humanoid Interface for ISAC

Physical aspects are those that pertain to the structure or body of the human and the humanoid. These aspects cover the essential physical features and manipulation capabilities of each.

Sensory aspects refer to the channels through which the human and humanoid gain information about the world and the means by which the two exchange information.

The Cognitive aspects describe those concerned with the internal workings; the mind of the human, and the condition of the humanoid. In addition to representing the emotional state of the human, such as in affective computing [17], we are interested in giving the humanoid its own emotional or affective system. Arkin [3] presents a discussion of the utility of representing emotion in robots.

To explicitly model these aspects in our humanoid system, we are developing a unified HHI framework (Figure 3). In this framework, the Human Agent encapsulates the aspects of the human, while the Self Agent handles the aspects of the humanoid. The Sensory EgoSphere (SES) and the DataBase Associative Memory (DBAM) are designed to provide the humanoid with the short-term and the long-term memory, respectively.

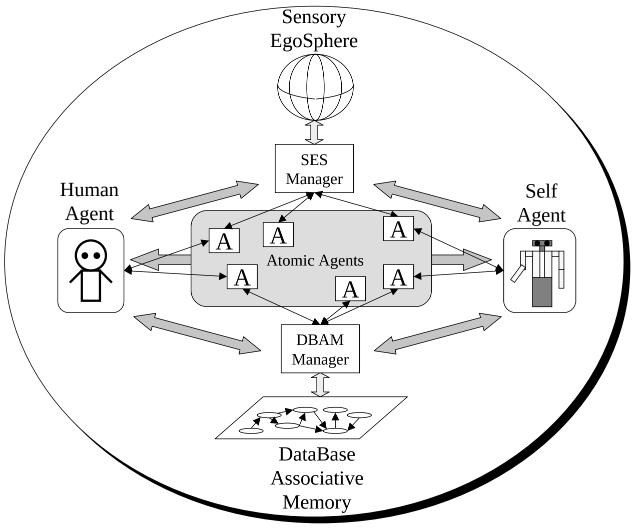

Fig. 3. A Framework for Human-Humanoid Interaction in IMA

3.1 The Human Agent

The Human Agent represents the human in HHI. The Human Agent is a compound agent composed of the atomic agents that are tuned to human features. The role of the Human Agent is to encapsulate the information that the humanoid has determined about the human; this lets the system interact intelligently with the human by sensing these features dynamically. The Human Agent detects, monitors, and identifies humans throughout its interaction. The current implementation of the Human Agent, shown in Figure 4, includes four modules logically grouped by function for detection, monitoring, identification, and interaction.

The Detection Module finds humans using visual tracking, infrared motion detection, and sound source localization [20]. The Monitor Module locates and tracks features of humans–currently faces and hands, thus covering the Physical aspects.

Other sensory aspects are covered by the Interaction Module, which monitors speech and gesture input. Its role is to pass speech commands to the Self Agent and to interpret gesture, thus allowing the human to supplement speech by deictic gesturing.

The Identification Module is used to identify individuals in the environment. This module monitors features, such as a person's height and clothing color information, to identify and detect changes among the people in its environment. Other features that will be exploited include speaker recognition and face identification.

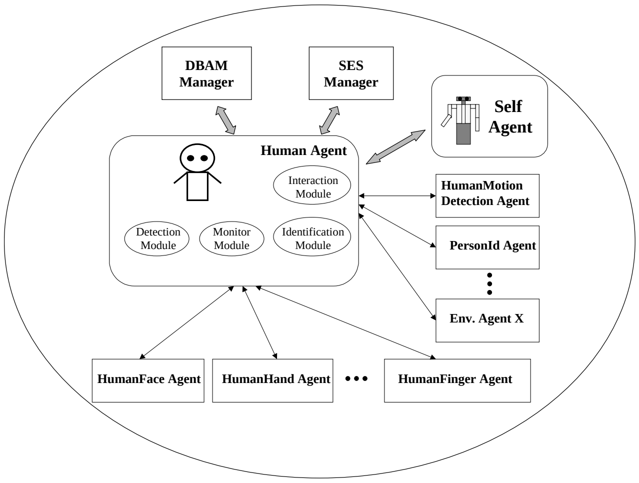

Fig. 4. Human Agent and its constituent modules in relation to other agents

3.2 The Self Agent

The Self Agent is a compound agent that addresses the cognitive aspects of the humanoid in HHI. The physical and sensory aspects are mapped to various atomic agents. Part of the cognitive aspect covered by the Self Agent is to activate sequencer and environment atomic agents in response to input from the Human Agent and the DBAM Manager. It also integrates failure information from atomic agents and maintains information about the overall state of the humanoid. The failure information is collected from System Status Evaluation, a low-level performance analyzer.

The Self Agent also generates part of the humanoid's communication to the human. The Self Agent does this in response to the human's input, acknowledging receipt of commands. It also takes initiative in generating output when the status information maintained by the Self Agent indicates a problem with the humanoid's functioning.

The current implementation of the Self Agent, shown in Figure 5, includes six modules. The Interaction Module handles input from the Human Agent and the DBAM Manager. This input represents speech phrases from the human and actions chosen by the spreading activation network in the DBAM manager (see Section 4.2 below). Input is processed by interpretation software based upon the work developed by Mauldin [13]; it works by matching input text against a limited number of patterns. The Activator Module is the means by which the Self Agent modifies the activation state of atomic agents. It maintains a list of atomic agents with associated links to each agent. The Interaction and Activator Modules work together to perform an action in response to a limited set of human inputs.

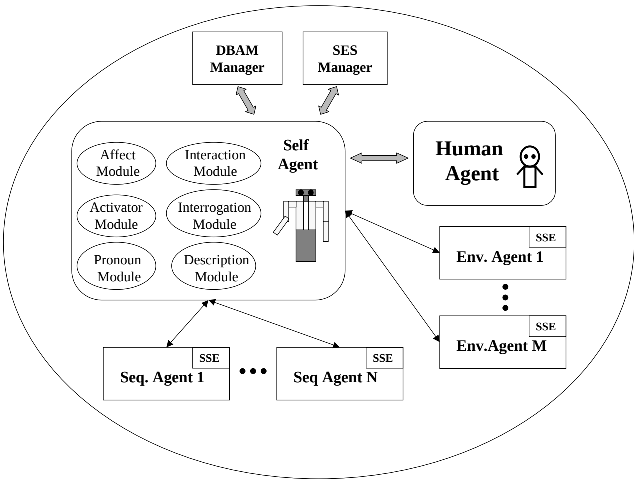

Fig. 5. Self Agent and its constituent modules in relation to the other agents.

The Affect Module is an artificial emotional model used to describe the current state of the humanoid similar to that described in [5]. Artificial emotions are modeled by numerical variables to produce a description of the humanoid's emotional state for feedback to the human. Currently, two artificial emotions are used. The first, *happiness*, reflects the operating status of the humanoid. Atomic agents which encounter a failure cause the happiness variable to become more negative. Another variable is *confusion*, which is modified by the Pronoun Module. The Pronoun Module provides conversational referents; that is, pointers to agents that represent the topic of discussion. Currently, the Pronoun Agent provides three pointers, the *subject*, the *object*, and the *task*. It binds environment atomic agents to symbolic names. The agents to be bound are determined from the input patterns processed by the Interaction Agent. This concept is similar to the *pronome* described in [14] and role-passing developed in [8]. If the Pronoun Module cannot resolve the correct binding—for example, the desired environment atomic agent may not exist—then the module will increase the confusion affect variable.

The Description Module maintains descriptions of the atomic agents available in the system. This allows agents to supply text strings containing their names and a description of their purpose. The atomic agents also supply information about their status: whether they are active or inactive, and whether they are working correctly or not. The Description Module then has text strings containing the names and descriptions of the atomic agents connected to it. These text strings can be used to provide feedback to the human, via the Interrogation Module. The Interrogation Module uses the same interpretation software as the Interaction Module; in this case, it matches input text agains patterns which represent a small set queries about the status of the humanoid. The responses to the queries are based on templates and use the descriptions of the agents in the Description Module.

3.3 System Status Evaluation

Humanoid robots are expected to have a rich interaction with humans in a wide range of situations. Therefore it is important for them to be able to recognize if they are working properly, and if not, to tell humans why.

One key feature of our agent-based system is *pervasive communication* among agents. Furthermore, there exist *patterns of data flow* with associated timing characteristics which must be maintained for correct operation. This idea has been used by Han and Shin to develop the heartbeat algorithm [7]. Some of the advantages of using communication timing information for failure detection are:

- Timing of communication is easily observed
- Timing patterns can be analyzed using existing signal processing techniques
- Analysis of timing patterns gives an indication of system status, which contributes to a "sense of health" for the humanoid system

Our strategy is based on the assumption that if an agent is unable to meet its goals (as defined by task-specific criteria) and is experiencing abnormal communication with another agent, this will indicate a problem. We have developed a low-level performance analyzer called System Status Evaluation (SSE) for the statistics of agent communication timing, specifically, the delay between successive communication messages [2]. The advantage of this analyzer is that it uses observations local to each agent, instead of the entire system model, to detect failures.

Within IMA, a communication channel between two agents is called a *relationship*. The two most common relationships in our system are (1) *observer* and (2) *call-response*. In the observer relationship, one agent supplies another with a steady stream of data, often at regular intervals. As shown in Figure 6(a), agent S takes on the *source* role, sending messages to agent O, which takes on the *observer* role. We define the message stream as $M(k) = \{M(0), M(1), ...\}$. Figure 6(a) also shows the time delay between two successive messages $M(k-1)$ and $M(k)$, which we define as $\Delta(k) = t_k - t_{k-1}$.

In the call-response relationship, two agents act as a "call-response" pair. The agent in the caller role, C, sends the agent in the slave role, R, a signal to begin an operation. When the slave has completed its operation, it sends an acknowledgement signal. This is shown in Figure 6(b).

In this model, the caller sends a command $C(k)$; the responder, when finished with its task, sends a reply $R(k)$, where $t_{R(k)} > t_{C(k)}$. Figure 6(b) shows the time delay between a command and a corresponding reply, defined as $\Delta(k) = t_{R(k)} - t_{C(k)}$.

Agents which take on the observer role in an observer relationship, or agents which take on the caller role in a call-response relationship, can use the communication timing information to determine whether their corresponding observed

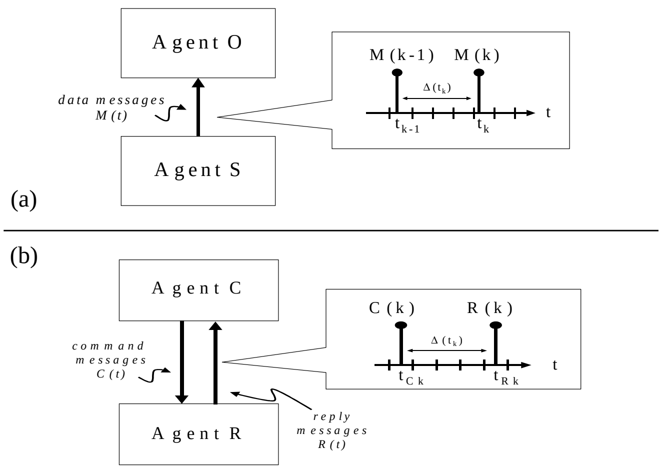

Fig. 6. Inter-Agent Communication: Observer and Call-Response Type

or slave agents have failed. These agents measure Δ during normal operation and create a histogram of the values. Figure 7 shows an example of such a histogram, collected from an observer relationship involving the agent controlling one of ISAC's arms. This histogram has two spikes–a large one corresponding to a Δ of approximately 200ms, and a smaller one corresponding to approximately 300ms.

Using the histogram, an agent can estimate $F_{\Delta}(x; k)$, the probability that $\Delta(k) \leq x$.

Using the histogram, we can estimate $F_{\Delta}(x; k)$, the probability that $\Delta(k) \leq x$, by accumulating the histogram. If we designate the histogram by $H(k)$, where $0 \leq k \leq N$, then the function is estimated by $\frac{\sum_{n=0}^{k} H(n)}{\sum_{n=0}^{N} H(n)}$. Given this function, the agent can choose a probability threshold P_{max} that corresponds to the longest acceptable Δ, which we denote Δ_{max}, for that relationship. During operation, agents monitor the timing of their communication relationships. If an agent ever observes a communication delay greater than Δ_{max}, then it can classify the other agent in the relationship as failed. The agent can then report this failure to the Self Agent, which can inform the human.

This performance analysis has some limitations. To detect failure, SSE requires that the failure cause a change in communication timing. For example, any failure which completely interrupts the execution of the agent would be detectable. This kind of failure could be due to the host machine on which the agent runs, or due to software failures caused by programming errors. Likewise, any

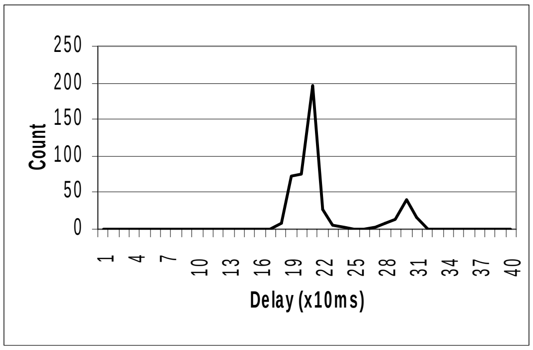

Fig. 7. Inter-Agent Communication Histogram

failure in communication channel—e.g., to the network infrastructure—would be detectable. The detectability of other types of failure depends on how the failure affects the agent's internal process that controls communication. For example, an agent may produce an incorrect output, but still meet communication timing limits; in this case, SSE would not detect a failure.

4 Memory Data Structures

4.1 Sensory EgoSphere

The Sensory Egosphere (SES) is a data structure that maintains a short-term memory of sensory data for the humanoid robot. The SES is comprised of a time-varying, spatially-indexed database and a geodesic hemisphere interface. The SES interacts with any IMA agent providing or needing short-term sensory data using the SES Manager agent. This Manager facilitates communication between the SES and other IMA agents so that sensory data can be registered onto and retrieved from the SES. Data is registered onto the SES using the location of the humanoids head. Not all sensory data in the environment is registered. Registration occurs only for data that is relevant to the humanoid's actions. Also, registered data is stored at nodes on the SES. These nodes are spatially-indexed according to the pan and tilt angles of the humanoid's head. Data is stored at the closest node to its registration location. Data is retrieved from the SES using either the data name, the data modality or a desired spatial location. The retrieval uses a robust search of the SES according to the method chosen. If the data name or modality is used for retrieval, then the entire SES is searched. If a desired spatial location is used for retrieval, the search is restricted to that location, but all data at the location is retrieved.

Albus originally conceived the egosphere and defines it as "a spherical coordinate system with the self (ego) at the origin" [1]. Our implementation uses a geodesic hemisphere as a coordinate system. The hemisphere is centered at the humanoids head and registers data occurring in the frontal view of the robot. The hemisphere stores data in a topological fashion. The spatial information on the hemisphere is of two dimensions: azimuth and elevation. Data in the humanoids environment is "projected" onto the SES according to these dimensions. Figure 8 shows this topological orientation using objects in the environment.

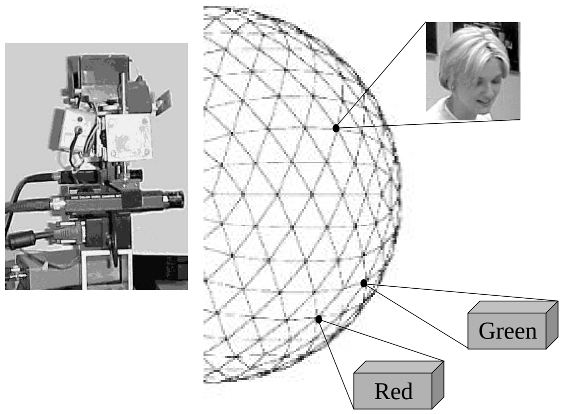

Fig. 8. Projection of Data onto SES

The geodesic hemisphere is composed of triangles that can have different tessellations. A larger tessellation gives a better resolution of the SES. Therefore, data of a coarse granularity can be stored on the SES at one tesselation, while data of a finer granularity is stored at a larger tessellation. This tessellation creates neighborhoods of different depths. These neighborhoods construct the topology of the SES. When data is being retrieved from the SES, neighborhoods of nodes are searched. The vertices of the triangles on the geodesic hemisphere represent the nodes of the SES.

Figure 9 shows an example of retrieving data from the SES using a desired spatial location. This search begins at the node closest to the desired location, which in Figure 9 is the point A. In this example, a neighborhood depth of 1 is given for the search. This depth indicates how many neighborhoods around the beginning node will be searched. The SES searches for all registered data beginning at the initial node. The search directions are denoted as arrows in the

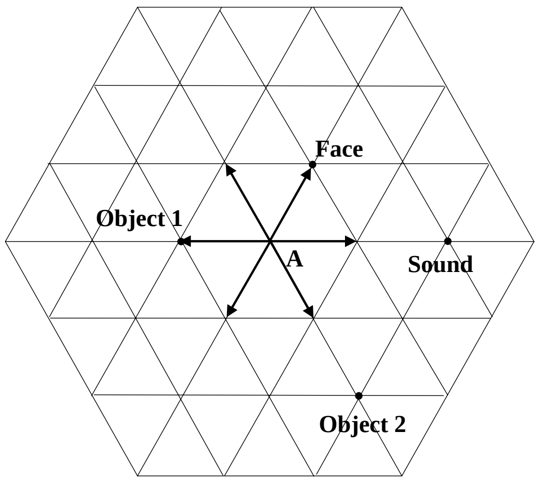

Fig. 9. Retrieval of Data Using a Desired Location

figure. The search continues to those nodes in a neighborhood depth of one. The search is complete when the nodes in the final neighborhood depth have been searched. In this example, the search is complete after the first neighborhood has been searched. The SES relays all retrieved data to its manager. For this search example, the data retrieved is "Face" and "Object 1". If the neighborhood depth is given as two, then the data retrieved includes "Face", "Object 1", "Object 2" and "Sound".

To retrieve data using a data name or modality, the search begins in the center of the SES. Every node on the SES is searched and all data matching the specifications is retrieved. The SES then sends this information to its manager.

4.2 Database Associative Memory

The DataBase Associative Memory (DBAM) is a data structure that provides a long-term memory for the humanoid robot. The DBAM allows the humanoid to recall sequences of actions based on the state of its internal and external environments. The DBAM is comprised of a database for holding memory information and a probabilistic Spreading Activation Network (SAN) for performing memory recall [4]. The DBAM interfaces with a DBAM Manager agent for communication with other IMA agents. The DBAM Manager allows other agents to cue memory, modify memory and add to memory.

The SAN operates on action nodes and condition nodes to determine the next action the humanoid should perform. The action nodes describe various

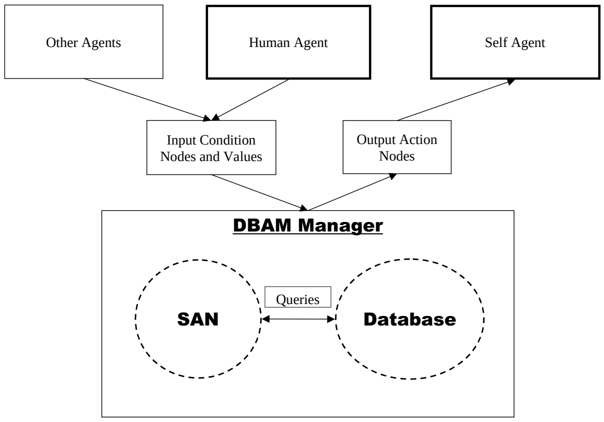

Fig. 10. DBAM Communication

levels of tasks. One action node may contain the name of a sequencer agent that performs multiple tasks while another action node may contain the name of a skill agent that performs one task. The condition nodes represent the state of the humanoid, both internally and externally. These nodes provide the associations between action nodes. Condition nodes represent either conditions that should be met for an action to be performed (pre-conditions) or conditions that are met by an action that has been performed (post-conditions). The condition nodes are valued according to the current state of the humanoid and to the desired goal state. The purpose of the SAN is to achieve the desired goal state through the shortest path of associated actions. Since the SAN uses a probabilistic algorithm, the humanoid is capable of reactive planning. However, this SAN also allows for deliberative planning and for learning.

The database holds records containing the action nodes, condition nodes and probabilities pertaining to the nodes and their associations. All information stored in the database is utilized in the SAN. The SAN communicates with the database directly by querying for retrieval of information. The database can be accessed by other IMA agents using the DBAM Manager. This allows other agents to update records and to add new records within the database.

The DBAM is a data structure, not an agent. The DBAM does depend on communication with other agents. Figure 10 shows the communication between the DBAM and other agents and the communication within the DBAM.

While the DBAM currently uses a single-layer network, the SAN can be expanded to include multiple layers (Figure 11). In the multi-layer SAN, action

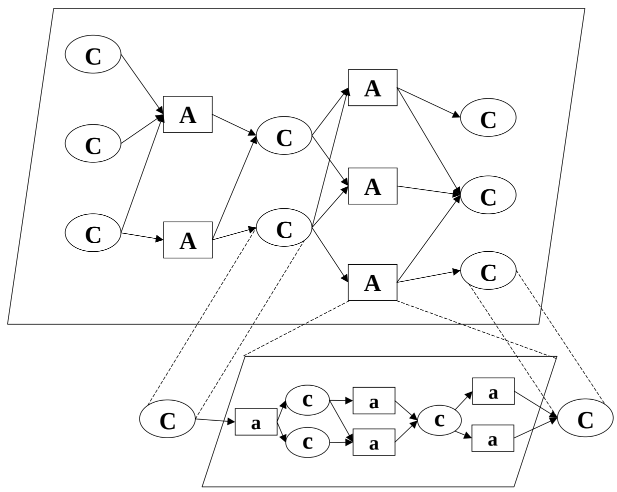

Fig. 11. A Multi-Layered Spreading Activation Network

nodes on any layer can encapsulate other networks. As our system becomes more complex, the single-layer SAN will only expand outward. With a multi-layer network, we hope to modularize SANs, so that more actions could be added easily.

5 Current Applications

5.1 A Handshaking Demonstration

We have developed a demonstration in which ISAC can shake hands with a person (Figure 12). In this demonstration, ISAC interacts with a human in several ways. Each atomic agent in the demonstration adheres to a simple script, but the interplay among atomic agents and with the human provides for a variety of situations. Figure 13 shows the interconnections among some atomic agents in the system.

A typical scenario begins when the Human Agent detects the presence of a human and begins tracking the face. If the Human Agent cannot identify the human, it will send a message to the DBAM Manager indicating that the person is unknown, and that the goal state is "Greeting Complete" (Figure 14(a)). The DBAM Manager then sets the value of the "Person Unknown" condition node to True and the utility of the "Greeting Complete" node to 1.0. The spreading activation network then selects the "Greet and Introduce" action. The DBAM Manager sends this information to the Self Agent, which produces the speech output, "Hello, I am ISAC."

Fig. 12. ISAC Shaking Hands with a Person

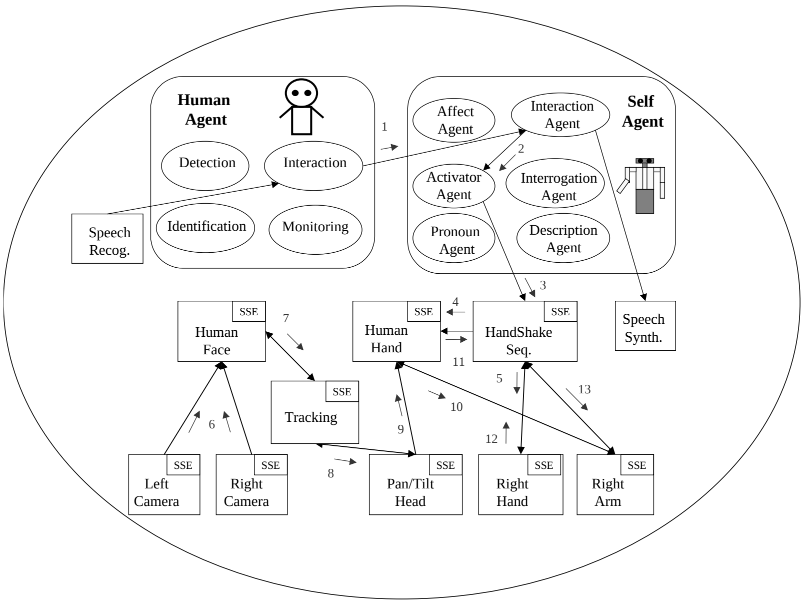

Fig. 13. Atomic Agents for the Handshaking Demonstration and their Relation to Human and Self Agents

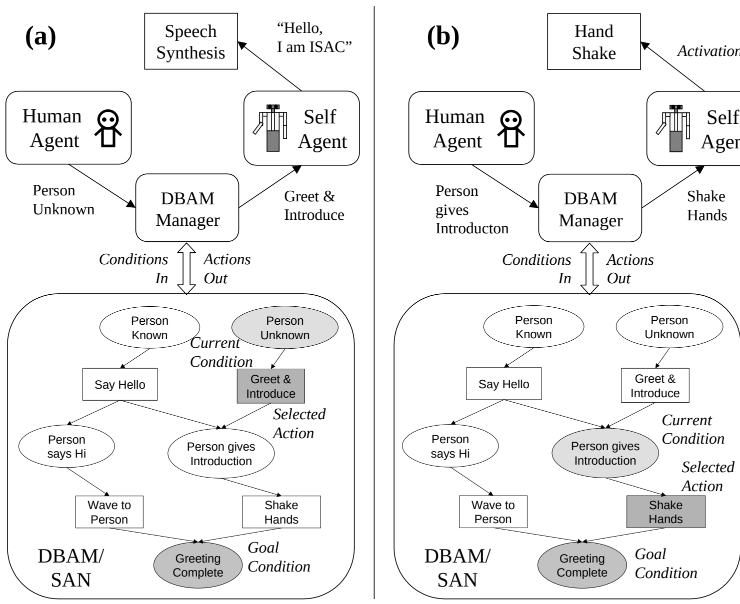

Fig. 14. Information Flow through the Framework During Handshaking

If the human then introduces herself or himself, the Human Agent sends a message to the DBAM manager, which sets the value of the "Person gives Introduction" node to True (Figure 14(b)). The spreading activation network selects the "Shake Hands" action, and the DBAM manager informs the Self Agent. The Self Agent will then activate the HandShake sequencer atomic agent. If the human is standing outside ISAC's workspace, ISAC will ask the human to come closer before attempting to shake hands.

5.2 A Finger-pointing Demonstration

The finger-pointing demonstration (Figure 15) utilizes the HHI framework to direct ISAC's attention to objects in its workspace. The application involves several levels of the framework for directing attention to known and unknown objects. To direct ISAC's attention to an unknown object, the application begins with speech direction from the human. This direction informs ISAC to look at a specific object. The Human Agent translates the speech command into a text command, which initiates the Find Finger agent. At the same time, the Human Agent stores the name of the object. The text command informs the Find Finger agent to find a pointed finger and to fixate on the object to which it is pointing. The Find Finger agent also retrieves the location of the object and returns the location to the Human Agent. After ISAC has fixated on the object, the

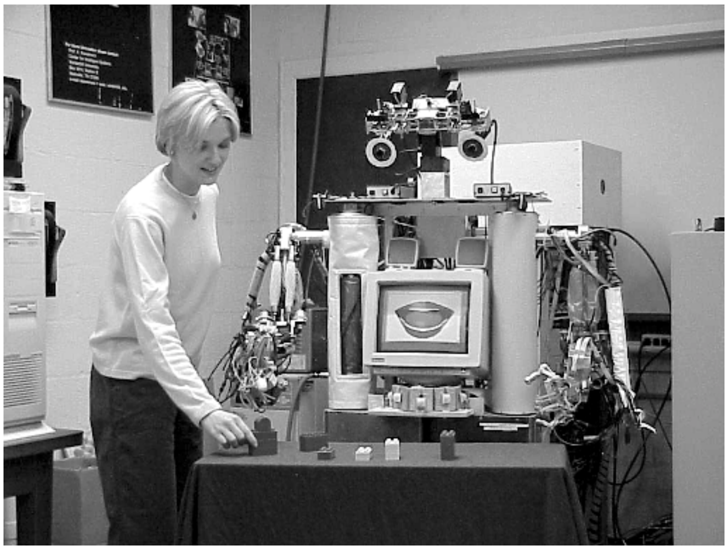

Fig. 15. Finger-Pointing

Human Agent sends the object name and location to the SES Manager. The SES Manager registers this object on the SES.

To direct ISAC's attention to a known object that is registered on the SES, the human tells ISAC to find the desired object. The Human Agent translates this speech command into a text command that sends the object name to the SES Manager. The SES Manager initiates a search of the SES using the object name. When the object is retrieved, the SES Manager returns the location of the object to the Human Agent. The Human Agent then sends the location to the Saccade Agent which directs ISAC to look at the object.

Figure 16 shows the HHI framework used to perform this application.

In the current application, ISAC is directed by a human to look at one of two objects. The objects are located in ISAC's workspace which is a table directly in front of the robot. The two objects are a red block and a green block. When ISAC is told to look at the green block, ISAC saccades to the pointed finger [21]. ISAC then takes the position of the green block from the location of the pointed finger. ISAC registers the location and name of the green block onto the SES. ISAC is directed to look at the red block and repeats the above actions. AFter the blocks are registered onto the SES, ISAC returns to its initial position (looking straight ahead). Then ISAC is told to look at either the red block or the green block. ISAC retrieves the object named from the SES. ISAC saccades to the location given in the SES.

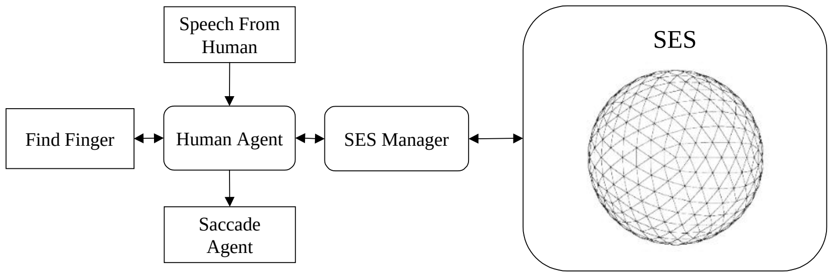

Fig. 16. The Flow of Information through the Framework during Fingerpointing

6 Conclusion

In this paper, we presented our approach for a unified framework for efficient and effective human-humanoid interaction (HHI). The HHI framework shown in Figure 3 has been developed on an agent-based architecture called the Intelligent Machine Architecture (IMA) using the ISAC humanoid system.

Our next step is to implement task learning mechanisms. We call our approach *learning sensory-motor coordination* (SMC). SMC is fundamental to the development of robot intelligence [16]. It provides the basis for physical control over objects. To achieve this goal, we are developing a control system that will enable ISAC to learn SMC while being led repeatedly by a person through a sequence of behaviors to perform a task [6]. This, we believe, will lead to the successful integration of humanoid robots into society.

References

1. J.S. Albus. Outline for a theory of intelligence. *IEEE Trans. Systems, Man, and Cybernetics*, 21(3):473–509, 1991.
2. A. Alford, M. Wilkes, and K. Kawamura. System status evaluation: Monitoring the state of agents in a humanoid system. To appear in *Proceedings of the 2000 International Conference on System, Man, and Cybernetics*, October 2000.
3. R. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.
4. S. Bagchi, G. Biswas, and K. Kawamura. Interactive task planning under uncertainty and goal changes. *Robotics and Autonomous Systems*, 18(1-2):157–167, July 1996.
5. C. Breazeal(Ferrell). Regulating human-robot interaction using "emotions", "drives" and facial expressions. In *Autonomous Agents 1998 workshop "Agents in Interaction - Acquiring Competence through Imitation"*, 1998.
6. M. Cambron and R. A. Peters II. Sensory motor control for grasping in a humanoid robot. To appear in *Proceedings of the 2000 IEEE International Conference on Systems, Man and Cybernetics*, October 2000.

7. S. Han and K. G. Shin. Experimental evaluation of failure-detection schemes in real-time communication networks. *IEEE Transactions on Parallel and Distributed Systems*, 10(6):613–626, June 1999.
8. I. Horswill. Real-time control of attention and behavior in a logical framework. In *Proceedings of the First International Conference on Autonomous Agents*, pages 130–137, February 1997.
9. N. Jennings, K. Sycara, and M. Woolridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, (1):7–38, 1998.
10. K. Kawamura, R.A. Peters II, S. Bagchi, M. Iskarous, and M. Bishay. Intelligent robotic systems in service of the disabled. *IEEE Transactions on Rehabilitation Engineering*, 3(1):14–21, March 1995.
11. K. Kawamura, R.A. Peters II, D.M. Wilkes, W.A. Alford, and T.E. Rogers. Isac: Foundations in human-humanoid interaction. To appear in *IEEE Intelligent Systems*, July/August 2000.
12. K. Kawamura, R. T. Pack, M. Bishay, and M. Iskarous. Design philosophy for service robots. In *Robotics and Autonomous Systems*, volume 18, pages 109–116. July 1996.
13. M. Mauldin. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 16–21, Seattle, WA, August 1994.
14. M. Minsky. *The Society of Mind*. Simon and Schuster, Inc., New York, NY, 1985.
15. R.T. Pack, D. M. Wilkes, G. Biswas, and K. Kawamura. Intelligent machine architecture for object-based system integration. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Machatronics*, pages 3774–3779, Tokyo, Japan, 1997.
16. R. Pfeifer and C. Scheier. *Understanding Intelligence*. The MIT Press, Cambridge, MA, 1999.
17. R. Picard. *Affective Computing*. MIT Press, 1997.
18. G. Richter, F. Smieja, and U. Beyer. Integrative architecture of the autonomous hand-eye robot JANUS. In *Proceedings of CIRA'97*, Monterey, CA, July 1997.
19. D. Rogerson. *Inside COM: Microsoft's Component Object Model*. Microsoft Press, 1997.
20. A. Sekmen. *Human-Robot Interaction Methodology*. PhD Dissertation, Vanderbilt University, August 2000.
21. A. Srikaew. *A Biologically Inspired Active Vision Gaze Controller*. PhD Dissertation, Vanderbilt University, August 2000.
22. D. M. Wilkes, W. A. Alford, M. Cambron, T. Rogers, R. A. Peters, and K. Kawamura. Designing for human-robot symbiosis. *Industrial Robot*, 26(1):49–58, 1999.

Acknowledgements

This work has been partially funded by the
DARPA Mobile Autonomous Robotics Systems (MARS) Program.