

# Implementation of Cognitive Control for Robots

**Kazuhiko Kawamura, Stephen M. Gordon, Palis Ratanaswasd, Christopher S. Garber,  
and Erdem Erdemir**

Center for Intelligent Systems

Vanderbilt University

Nashville TN 37235-0131 USA

{kaz.kawamura;stephen.m.gordon;palis.ratanaswasd;chris.garber;erdem.erdemir}@vanderbilt.edu

## ABSTRACT

Engineers have long used control systems utilizing models and feedback loops to control real-world systems. Limitations of model-based control led to a generation of intelligent control techniques such as adaptive and fuzzy control. Human brain, on the other hand, is known to process a variety of inputs in parallel, ignore distractions to focus on the task in hand. This process, known as *cognitive control* in psychology, is unique to humans and some higher-class animals. We are interested in implementing such cognitive control functionality in robots. This paper tries to answer the following question: *How could cognitive control functionality be implemented in HAM-inspired robots?*

**KEYWORDS:** intelligent control, cognitive control, behavior learning, working memory

## 1. INTRODUCTION

As the need to control complex systems increases, it is important to look beyond engineering and computer science for new ways to control robots. For example, humans have the capacity to receive and process enormous amount of sensory information from the environment, exhibiting integrated sensorimotor associations as early as two years old [1]. A good example of such sensorimotor intelligence by adults is the well-known Stroop test [2]. Most goal-oriented robots currently perform only those or similar tasks they were programmed for and very little emerging behaviors are exhibited. What is needed is an alternative paradigm for behavior learning and task execution. Specifically, we see cognitive flexibility and adaptability in the brain as desirable design goals for the next generation of intelligent robots.

At the HAM Workshop in 2006, a concept of human cognitive control [3] and a multi-agent-based, hybrid cognitive architecture for robots [4] were presented. In this paper, we will present the progress made during the last year on the cognitive architecture and control, working memory training, and a self-motivated, internal state-based action selection mechanism.

## 2. COGNITIVE CONTROL FOR ROBOTS

Engineers have long used control systems utilizing feedback loops to control mechanical systems. Figure 1 illustrates a class of adaptive/learning control systems [5]. Limitations of model-based control led to a generation of *intelligent control techniques* such as fuzzy control, neuro computing and reconfigurable control.

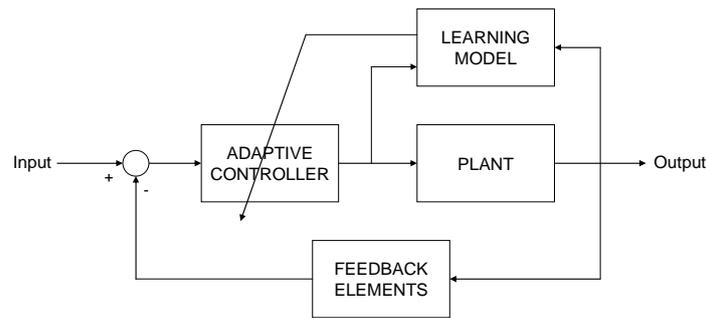


Figure 1. An Adaptive Control System [5]

The human brain is known to process a variety of stimuli in parallel, ignore non-critical stimuli to execute the task in hand, and learn new tasks with minimum assistance. This process, known as *executive* or *cognitive control*, is unique to humans and a handful of animals [6]. Figure 2 illustrates a conceptual model of cognitive control which we are using to realize robust behavior generation and learning for our humanoid robot.

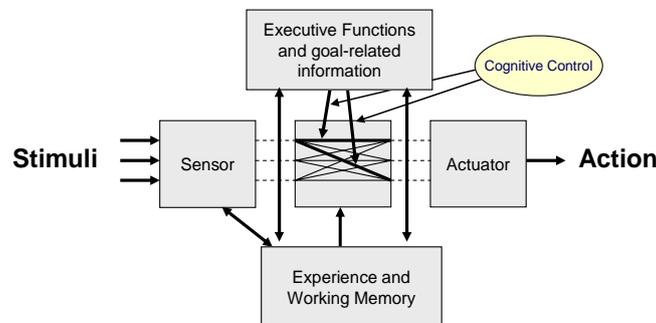


Figure 2. Model of Cognitive Control Modified from Miller, et. al [6]

As the complexity of a task grows, so do the software complexities necessary to process sensory information and to control actions purposefully. Development and maintenance of complex or large-scale software systems can benefit from domain-specific guidelines that promote code reuse and integration through software agents. Information processing in our humanoid robot ISAC (Intelligent SoftArm Control) is integrated into a multiagent-based software architecture based on the Intelligent Machine Architecture (IMA) [7]. IMA is designed to provide guidelines for modular design and allows for the development of subsystems from perception modeling to behavior control through the collections of IMA agents and associated memories, as shown in Figure 3.

For any learning system, memory plays an important role. As Gazzaniga et al., states, “Learning has an outcome, and we refer to that as *memory*. To put it another way, learning happens when a memory is created or is strengthened by repetition.” [1, p. 302]. ISAC’s memory structure is divided into three classes: Short-term memory (STM), long-term memory (LTM), and the working memory system (WMS). STM holds sensory information of the current environment in which ISAC is situated. LTM holds learned behaviors, semantic knowledge, and past experience. WMS holds task-specific information called “chunks” and streamlines the information flow to the cognitive processes during the task execution. STM is implemented using a sparse sensory data structure called the Sensory EgoSphere (SES). It was inspired by the egosphere concept defined by Albus [8] and serves as a spatio-temporal STM [9]. LTM stores information such as *skills learned* and *experience gained* for future recall.

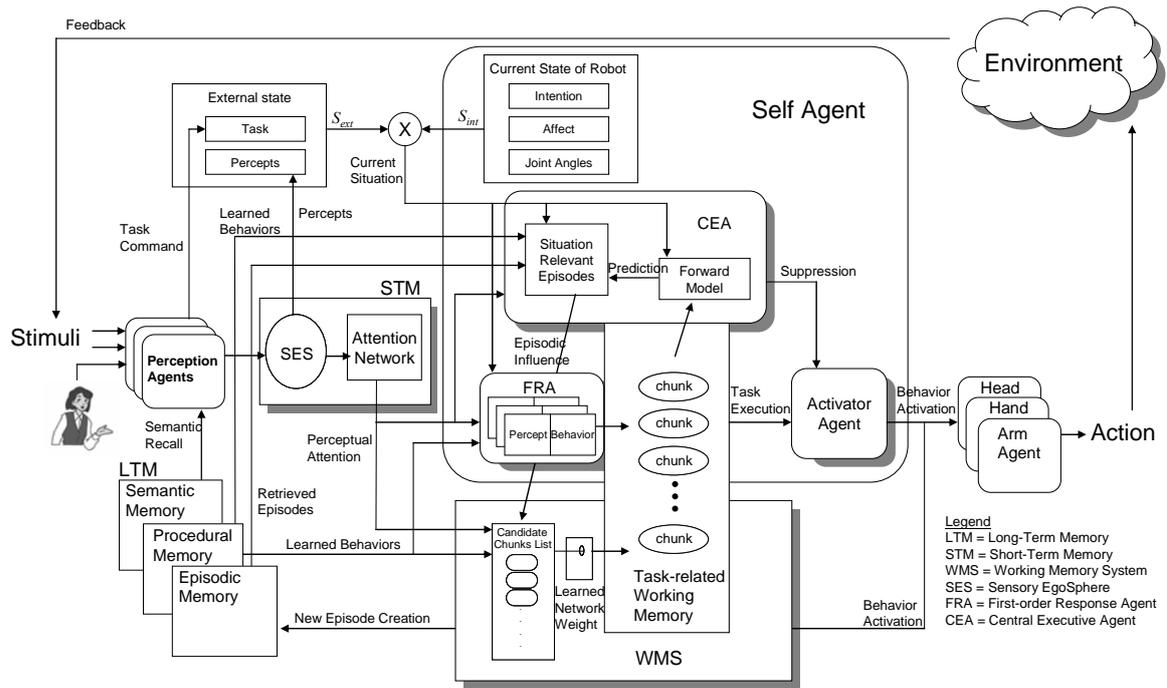


Figure 3. Multiagent-Based Cognitive Robot Architecture

### 3. THE WORKING MEMORY SYSTEM

#### 3.1 Central Executive and Working Memory System

Cognitive functions ISAC can perform, i.e. cognitive control, are modeled after Baddeley’s human working memory model [10]. In his model, the “central executive” controls two “working memory” systems: the phonological loop and the visuo-spatial sketch pad. Cognitive control functions are currently implemented using STM, LTM, the Attention Network and WMS. As discussed, STM handles sensor-based percepts. These percepts can be assigned *focus of attention* (FOA) or *gating* by the Attention Network [11]. This happens as a result of associated knowledge (such as emotional salience) with the sensed percepts. FOA-based percepts are then passed to the WMS as candidate task-related chunks.

“Biological working memory represents a limited-capacity store for retaining information over the short term and for performing mental operations on the contents of this store” [12]. This type of memory system is said to be closely tied to task learning and execution. [12]. Inspired by this, we have implemented the working memory structure into ISAC to provide the embodiment necessary for exploring the critical issues of task execution and learning. Our hypothesis is that this integration will lead to a more complex, but realistic robotic learning system involving perceptual systems, actuators, reasoning, attention, emotion, and short- and long-term memory structures.

#### 3.2 WM Training Experiments for Percept-Behavior Association Learning

Within the ISAC architecture, learning how to respond to novel tasks is done an untrained working memory system. When a novel stimulus is present, this system explores different responses and, over time, learns what information from short- and long-term memory should be focused on to best execute the novel task. As the system learns, the trained working memory (WM) is stored in episodic memory and can be retrieved in the future for quick, reliable, task execution. Experiments have been performed to utilize this task-learning portion of the ISAC cognitive architecture [13]. These experiments relate to tasks for which ISAC

had no previous experience with the situation at the time of training. Figure 4 shows the task learning loop involving the working memory within the cognitive architecture. Figure 5 shows sample configurations for the behaviors used in this section and the following sections.

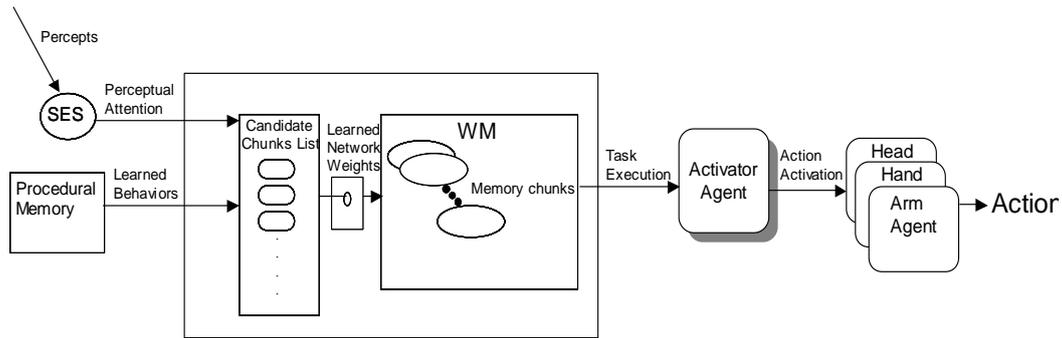


Figure 4. Task-Learning Loop in ISAC Cognitive Architecture

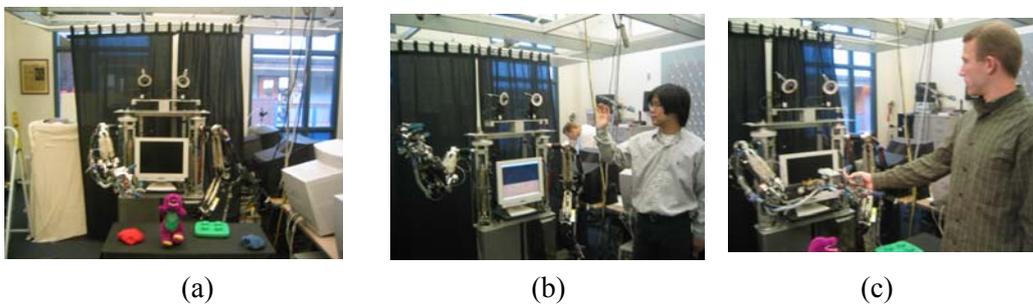


Figure 5. Sample Configurations for Behaviors used (a) Reach, (b) Wave, (c) Handshake

The training conducted utilizing the WMS represented initial trial-and-error responses to novel tasks. For each experiment, the WMS initialized an untrained instance of working memory. This was required because current computational limits only allowed a trained WM to be used for similar types of tasks. For instance, the tasks *reach to the bean bag* and *track the LEGO toy* are similar because each task requires one percept and one behavior. The task of interacting with a person, however, is not similar to these two and would require a separately trained WM. Novel tasks were those for which a trained WM could not be found. The interpolation and execution of behaviors in this experiment, and those described later, were performed using a modification of the Verbs and Adverbs algorithm [14], discussed further in Appendix 2.

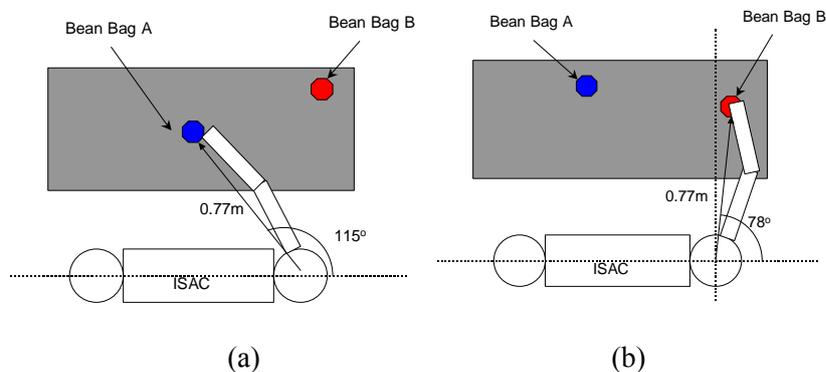


Figure 6. (a, b) Sample Configurations for Reaching

Thus far, WM has been successfully trained to perform the types of tasks discussed above, task involving one percept and one behavior. The example used in the remainder of this section will be, *reach to the bean bag*, as shown in Figure 6(a, b). In our trials, two bean bags were present and the WMS was required to choose one. Preliminary results for training this WM have been presented in [13].

During the initial training of this WM a high exploration percentage (15%) was used which helped avoid local maxima. In other words, 15% of the time the system continued to explore random actions even after a solution had been found. In addition, a reward rule was provided that rewarded the WM for chunk selection based on the success of the current trial. Once the WM began to converge on the appropriate responses, the other cognitive processes could then begin recording and storing episodes. However, currently a human trainer performed the action of deciding when to record episodes. Among the items stored in these episodes was the current WM. Therefore, when a similar situation is encountered in the future, ISAC could not only recall matching similar episodes but also retrieve the now-trained WM used during those episodes.

### 3.3 System Performance

The performance of the working memory during training was evaluated using that working memory's specified reward rules. During the learning process, reward was given based on the following three criteria:

1. Did the behavior chunk chosen successfully accomplish the task?
2. Did the percept chunk chosen successfully accomplish the task?
3. What was the difference between similar performances? (e.g. reaching to the nearest bean bag rather than the farthest one)

Reward criterion 3 was implemented to allow differentiation between similar choices. An example of this is the *reach to the bean bag* task represented in Figure 6. Note that two bean bags are present and a reach to either one would accomplish the task. However, it may be desirable to have ISAC understand that when the *reach to the bean bag* command is given, the intention of the instructor is actually to have ISAC reach to the nearest one, or perhaps always the blue or red one. Prior to task execution, the WMS had no understanding of this intention, but our experiments have shown that within approximately 20 trials (Figure 7 and 8), WMS learns this relationship. Furthermore, Reward criterion 3 can be changed without notice. When the WMS fails to receive reward when reward was expected, it began exploring alternative choices. In other words, it detected that the instructor's intention had changed and attempted to learn the new intention.

When the reward criteria were met, discrete positive reward was given to the system. No preference (i.e., reward of 0) was given if the system did not choose correctly. Implementing the exploration percentage encouraged exploration even after learning had been accomplished. This measure helped avoid local maximum.

Initial trials were performed in simulation to speed-up the testing phase of this percept-behavior learning. The simulation was removed the time-bottleneck of generating and performing behaviors. If the WMS desired to act on an object within the workspace, it was assumed that ISAC would be able to perform the desired action and reward was given accordingly. Appendix 3 shows the contents of short-term and long-term memory systems and some sample contents of working memory during training.

In these trials, WMS was allowed to choose two "chunks" from the short- and long-term memory systems to accomplish the task. However, the working memory was not restricted to choosing exactly one percept and one behavior. If the working memory chose to focus on two percepts, two behaviors, or chose not to load enough chunks then a behavior or percept was necessarily chosen at random. When there was not a behavior (or percept) chunk present, a random number generator was used to fill in the missing chunk. This ensured that an action was always performed. The reasoning for this was to encourage the WMS to make choices.

Without this safeguard, the WMS would begin avoiding the decision by not loading any chunks during the trials where the WMS was consistently making incorrect choices. This was a behavior inherent in the learning networks used to create the working memory system. Randomly filling in the blank chunks allowed the system to continue exploration during these trials.

To graphically demonstrate the ability of WMS better, training trials were also conducted that only required working memory to learn one chunk (percept or behavior) at a time. Figure 7 shows the learning curve for the behavior *reach* for the command *reach to the bean bag*. Prior to these trials, ISAC was taught a number of behaviors including three right arm behaviors, *reach*, *handshake*, and *wave*. Within 20 trials, the WMS learned to associate the command *reach* with the appropriate behavior. Figure 8 shows the same curve for learning the *bean bag* percept. Again, within 20 trials the WMS had learned not only to associate a bean bag with the command, but also that the *blue bean bag* was the intended bean bag. After 100 trials, the WMS quit receiving reward for the *blue bean bag*, the intention had changed to the *red bean bag* and within 20 more trials this intention had been learned.

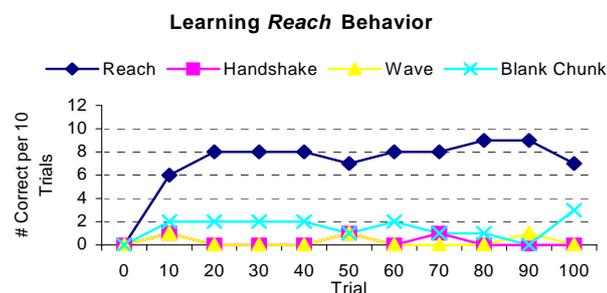


Figure 7. Learning to Associate the *Reach* Command with the Correct Behavior

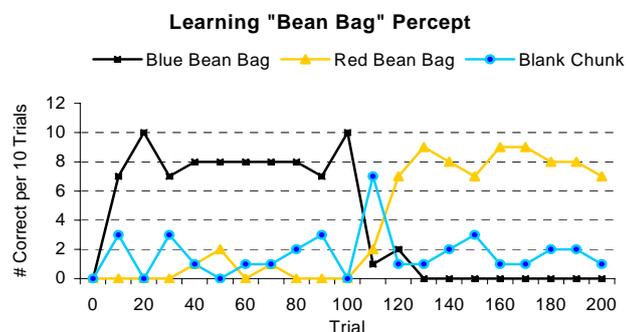


Figure 8. Learning to Choose the Correct *Bean Bag*.

## 4. THE ROLE OF CEA AND FRA FOR TASK SWITCHING

### 4.1 The First-order Response Agent (FRA) and the Central Executive Agent (CEA)

Figure 9 depicts the key IMA agents within the Self Agent. The Central Executive Agent (CEA) which is responsible for cognitive control during task execution invokes behaviors necessary for performing the given task. CEA operates in accord to *intention* which the Intention Agent interprets from a task command. Decision making in CEA is mediated by *affect* which is managed by the Affect Agent. The Activator Agent invokes head and arm agents to execute actions. The First-order Response Agent (FRA) is responsible for generating both routine and reactive responses. The term *first-order responses* refers to responses of the

system that are not generated from the cognitive process. This term was also used by Shanahan [15] in regard to responses generated reactively by the physical system which are contrast to responses generated by the higher-order loop which represent “imagination” in his work.

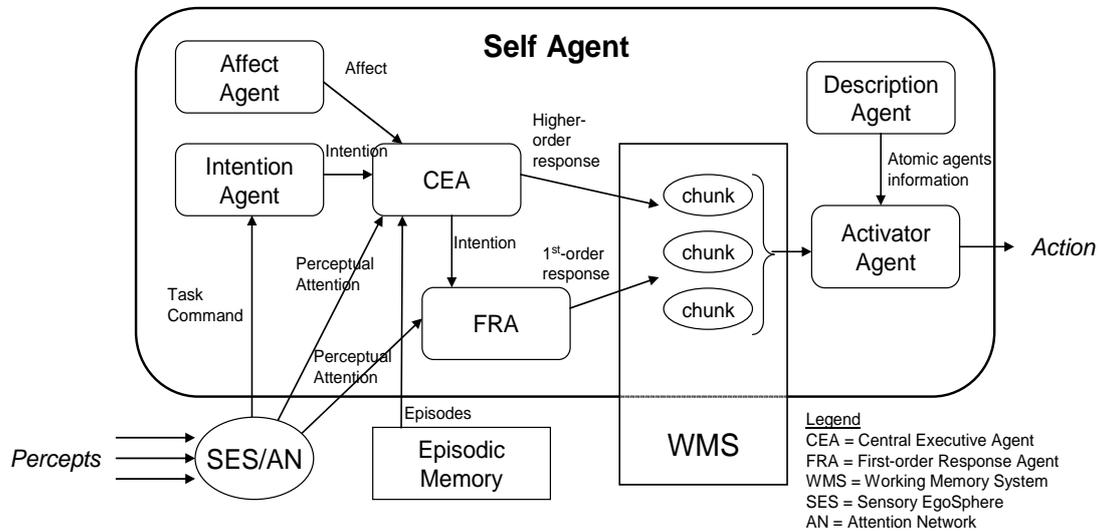


Figure 9. Structure of the Self Agent

First, reactive responses are handled by FRA by invoking corresponding behaviors when certain percepts receive ISAC’s attention. This concept is inspired by the schema theory where the system responds to certain stimuli by performing certain actions [16]. The associations between percept-behavior in FRA are provided as initial knowledge. FRA is implemented as a multithreaded process in which each stimulus-response pair is given its own separate running thread. As salient percepts on SES are put in Focus of Attention (FOA) by the Attention Network [11], each thread compares the systems current most salient percept with that particular threads percept from the percept-behavior pair. FRA posts both the matched percept and the behavior onto the Working Memory (WM) as chunks when a match is found. The Activator Agent then takes the chunks from WM and distributes to Atomic Agents in the system for behavior execution.

FRA has one thread that is responsible for routine responses. This thread invokes corresponding behaviors when certain situations are recognized according to the percepts in FOA and the current task. The recognized situation causes FRA to retrieve the learned skill associated with the situation from LTM. A learned skill contains the behavior needed to perform a particular task. Note that the current task could be assigned externally by a human or internally generated by self-motivation (see Section 5 for self-motivated decision making). FRA posts the behavior found in the retrieved learned skill and the percept in FOA into WM as chunks where the Activator Agent uses the chunks similarly to the case of reactive responses. However, the routine response thread will be subsumed when any one of reactive response threads are active. This phenomenon is similar to subsumption of behaviors in Brook’s Subsumption Architecture [17].

#### 4.2 FRA and Task Switching Experiment

A two-part experiment was conducted to validate how FRA can handle the routine and reactive responses. Figure 10 shows the IMA Agents and memory components utilized in the experiment.

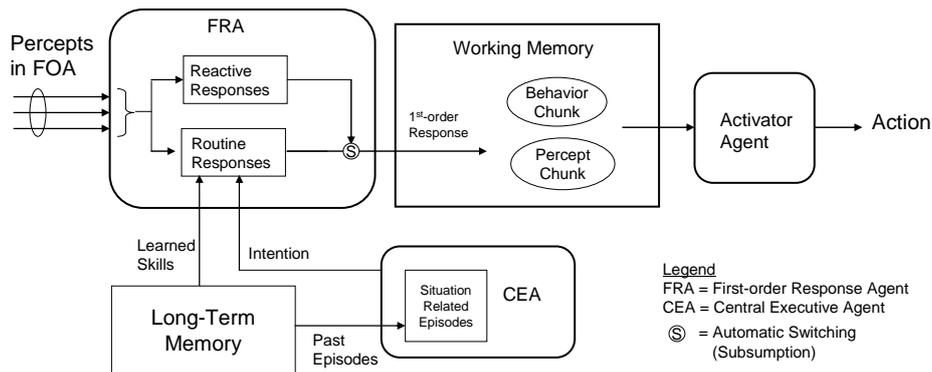


Figure 10. First-order Response in ISAC Cognitive Architecture

#### 4.2.1 Routine Response Experiment

The first part of the experiment was conducted to validate the capability to execute a task using a routine response and the ability to maintain the task context after a reactive response is invoked.

##### Experimental steps

1. ISAC actively monitors the environment.
2. Barney doll is placed within the field of view causing ISAC to recognize it, i.e., the situation, and to decide to play with the doll according to its innate knowledge.
3. When someone claps the hands, ISAC detects the location of the sound using the sound localization algorithm described in Appendix 4.
4. ISAC stops executing the task and saccades toward the source of the sound.
5. Because the task context is still active in the working memory, ISAC goes back to the task after the reactive response is completed.

#### 4.2.2 Task Switching Experiment

The second part of the experiment was conducted to validate the functionality of FRA to switch tasks when a new situation is recognized when an event occurs.

##### Experimental steps

1. ISAC continues the task from the above experiment.
2. Someone enters the room and approaches ISAC.
3. When a motion is detected at the door, using the motion detection method described in Appendix 4, ISAC stops executing the current task, fixates on the detected motion, and tracks the motion with the cameras.
4. When the motion enters the workspace, ISAC recalls a similar learned experience, thus executes the *handshake* behavior instead of going back to the previous task.

Figure 11 shows the lab view during the experiment.



Figure 11: Lab Views during Experiments - ISAC (a) Played with Barney, (b) Responded to Clapping Sound, (c) Detected Motion, (d) Shook Hands with the Person

This experiment shows a setup of ISAC cognitive architecture to perform simple cognitive control during task execution. In this setup, we focus on FRA which was used to execute task using first-order responses. To help providing simple cognitive control, CEA was used to generate task internally and make decision for task switching. In this section, we would like to evaluate FRA-based operation of the system using the following criteria

1. The ability to switch back and forth between reactive and routine responses
2. The ability to use routine responses to execute tasks
3. The ability to switch tasks based on situational change

In this experiment, the system has the ability to switch between reactive and routine responses seamlessly without losing its attention from the task. Reactive responses caused the system to immediately suspend the current task and attend to the particular percept that triggers the response. Reactive responses provided by FRA serve as a non-task oriented mechanism which helps the system to become aware of other events that happen in the environment that may require attention, therefore, the system should respond within a short period of time after an event happens. A short delay is expected, however, because of the complexity of the detection algorithms, and the propagation delay time in communication between agents. In the experiment, a set of clapping sounds were present at various angles where 0 degree was directly in front of ISAC. Table 1 summarizes the amount of time that the system took to respond after clapping sounds were heard, and the amount of time the system takes to resume the previous action after the reaction responses were completed.

Trial	Angles (degrees)	Response Time (ms)	Resuming Time (ms)
1	64.07	102	153
2	-38.65	906	143
3	18	105	135
4	-49	594	141
5	-2.65	716	139

Table 1. Response and Resuming Times to Loud Noise Reaction

Both parts of the experiment in this section show that FRA can be used to execute simple tasks successfully using knowledge about the task in LTM. Two tasks used in this experiment were *to play with Barney doll* and *to handshake the person*. Note that in this experiment, the tasks were not given by a person but instead were internally generated by CEA. The task information was passed between CEA and FRA using a shared memory slot. This method allows both agents to communicate very fast. FRA executed a task as soon as CEA posted the task to the memory slot.

The task switching experiment demonstrates the capability of the system to switch tasks based on situational change. The decision to switch tasks comes from CEA based on the situation. In this experiment, the system did not go back to the previous task because the robot detected that a person was moving toward it and entering the workspace. Due to strong association between the event and a task in the learned experience, CEA decided to switch to handshake with the person instead of going back to the previous task.

This performance evaluation in this section is performed based on partial results from the experiment. The final performance evaluation will be completed before the workshop.

## **5. SELF-MOTIVATED, INTERNAL STATE-BASED ACTION SELECTION MECHANISM**

Cognitive robots may face complex situations where they cannot rely on the state of the environment alone to make decisions. For example, the internal state of *affect* is shown to

play an important role in the human decision making [15] [18]. In our architecture, affect is considered a part of the internal state of the system. It is maintained by the Affect Agent, which keeps track of the current affective level of ISAC. Similar to the work of Shanahan [15], affect interacts with CEA by running in parallel, influencing focus of attention, and offering mediating task execution by influencing the probabilistic decision making model in the CEA [19]. However, unlike Shanahan’s work, affect in our architecture does not offer executive veto power. This veto power is kept by CEA.

The total state of the system is represented by two sets of state variables, *external* and *internal*. External state variables are represented by percepts and are placed in the Focus of Attention. Internal state variables include ISAC hardware parameters such as joint angle positions as well as variables such as intention and affect (Figure 3). These variables,  $S_{ext}$  and  $S_{int}$ , combine to form the overall situation.

$$Situation (S_{total}) = S_{ext} \times S_{int} \quad (1)$$

Only a portion  $S_{total}$  is required for the work discussed in this section. The internal state variable used is  $S_{affect}$ . The important external variables are those represented by  $S_{percepts}$  and  $S_{task}$ . As will be discussed below,  $S_{affect}$  is going to be determined quantitatively from these external variables.

In the ISAC cognitive architecture, task switching is based on the appearance of *events*, time-critical salient changes in the external state. The response for the current event depends not only on the current situation and the internal state, but also on the past experiences ISAC has encountered. The past experience of an event is stored as an “episode” within the Episodic Memory. We call the ability for ISAC to make decisions using the current state variables and past experiences (episodes) “situation-based action selection” (Figure 12).

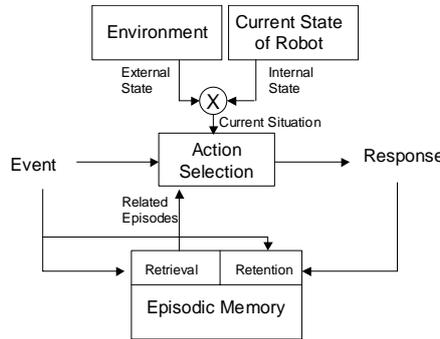


Figure 12. Situation-Based Action Selection Mechanism

## 5.1 Affect Agent and Excitement Meter

Robots operating in the real world often encounter situations where more than one course of action could be considered appropriate. When ISAC encounters a situation for which two separate episodes are retrieved from the episodic memory, where each episode involves a different response to the situation, it is necessary to choose one in order to continue task execution. In any cognitive robot, this choice should not be made based on hard-coded *if-then* rules. Rather it should be *mediated* by past experiences and internal motivations. This mediation enables ISAC to make its own choices to deal with competing situations. Towards the goal of internally motivated task mediation, we have begun developing a means of allowing ISAC to make self-motivated decisions based on its own preference, or the affect [18]. In our architecture the Affect Agent determines affectual associations with the current situation and provides suggestions to the CEA that impact decision-making. The suggestions made to the CEA indicate which choices would lead to higher or lower affectual states. The suggestions cause an increase (or decrease) in the probability that a particular course of action is chosen. In other words, the Affect Agent tells the CEA to increase (or decrease) the

probability that an action is chosen. Because the CEA can ignore the input from the Affect Agent, this input is regarded as a suggestion.

The CEA system implements a probabilistic model when making decisions with conflicting goals [19]. Past experience from episodic memory is used to fill in these probabilities. When episodes are retrieved, a list of possible actions is created and each action is assigned a priority,  $p_j$ . The probability that action,  $A_j$  is chosen is calculated as the priority of that action divided by the sum of the priorities of all actions. Further details of this probabilistic model are discussed in Appendix 1. The Affect Agent modifies these probabilities by using the affect associated with a particular set of stimuli to proportionately change the priority associated with that set of stimuli or action. By updating this priority, the probability that action is chosen is increased (or decreased). For example, if two sets of stimuli are present, the probability that the stimulus with the highest associated affect is chosen is increased. Likewise, the probability of choosing the second stimulus (the least affectual stimulus) is decreased.

In Figure 13, the current situation is input into the Affect Agent which calculates a new value for excitement and feeds this value back into the internal state. The current situation is passed on to the CEA, which uses that information to retrieve similar episodes from episodic memory and create an action list. Probabilities are assigned to the actions in the action list and these probabilities are updated by the affect from the Affect Agent.

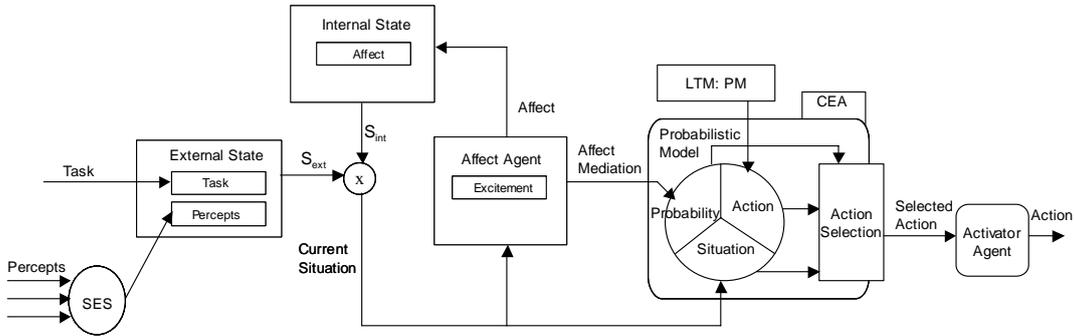


Figure 13. The Role of the Affect Agent in the ISAC Architecture

Prior work involving the Affect Agent [19] used pre-determined, fixed affect vectors. It is important to note that there are many potential affective state variables for ISAC but the current model will focus on only a single affect variable *excitement*. Our model of the Affect Agent (Figure 13) uses the following function for determining the affective state proposed by Anderson [20] but has also been suggested by Picard [18]

$$Excitement = Ae^{-Bt} \quad (2)$$

where

$$A = f(S_{ext}) \quad (3)$$

$$B = g(S_{ext}) \quad (4)$$

The values of A and B are parameters that are determined by the external state,  $S_{ext}$ . For example, situations associated with the action *reach to the bean bag* may retrieve a low value for A but a high value for B, indicating that these situations are not very exciting and that they decay rapidly. The variables (A, B) are functions of ISAC's current state, but they also represent one level of learning within this model. For instance, in a given situation the values of (A, B) can be modified to relate that the particular situation should no longer be deemed as exciting or, in fact, is to be considered more exciting the next time it is encountered. This is done by increasing or decreasing the stored values of (A, B) for particular situations.

Equation (2) assumes that ISAC's excitement level is continuous. As discussed above, task switching is based on the appearance of *events*, therefore it is more appropriate to use an

*event-based hybrid system* [21] to calculate affect. For example, the action *reaching to the bean bag* is a continuous action, however the *appearance (or disappearance)* of the bean bag percept is a discrete event. In order for the Affect Agent to deal with this, a second system (whose input is the external state and whose output is a discrete affect variable – D) is used as a switching mechanism from one situation to another, as shown in Figure 14. This switching mechanism re-initializes the local, event-based time used by the Affect Agent.

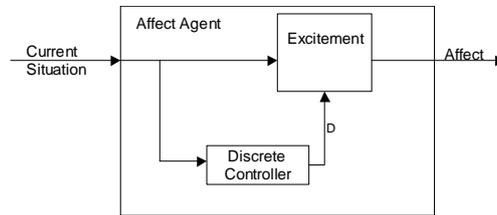


Figure 14. Hybrid Structure of the Affect Agent

Using our computational model of affect, the ISAC architecture is designed to mediate task switching to favor more exciting situations. The current implementation within the Affect Agent for the variable *excitement* is termed the *Excitement Meter*. This meter calculates the excitement response level to sets of stimuli and displays the history of excitement value graphically on the “chest” monitor mounted on ISAC (Figure 15). The current value of excitement is shown on the black axis. The previous values trail off to the right. Figure 15(a) shows the chest monitor as it normally appears. Figure 15(b) shows a close-up of the chest monitor when the excitement meter is displayed. Two jumps in excitement can be seen in the history section of the meter.

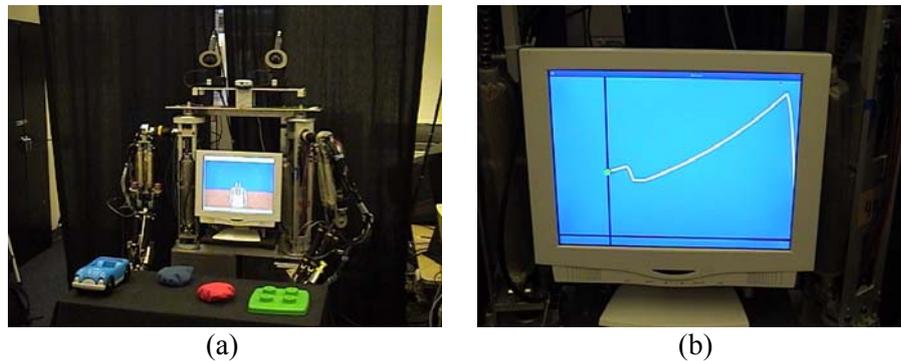


Figure 15. (a, b) ISAC Chest Monitor

## 5.2 Excitement Meter Experiment

An experiment involving the *Excitement Meter* has been designed to validate the functionality of the Affect Agent and to demonstrate the ability of this system when using affect in cognitive decision-making. In the designed experiment, competing sets of stimuli are presented to ISAC. The first set of stimuli is a command (from a human) to perform a task. The second set of stimuli is a number of toys that ISAC enjoys playing with. Each set of stimuli will have an associated level of excitement, and this excitement level (along with past experiences and the current situation) will help CEA to choose what to focus on in order to either (a) perform the task on hand, or (b) play with the toys.

### Experimental Steps

1. A pair of toys that ISAC can recognize are placed on a table in front of ISAC.
2. ISAC recalls a past episode which involved playing with the toys and subsequently begins to play with the toys.
3. A person enters the room and ask ISAC to perform a task. The task is encoded using key word search and posted onto SES.

4. CEA recognizes this new stimulus (i.e., task command) and the Excitement Meter calculates excitement associated with this situation.
5. The Excitement Meter passes the current excitement associations to CEA, where the decision is made to switch the task or not.
6. Based on the decision by the CEA, ISAC selects the appropriate action.

### 5.3 System Performance

Unlike many physical systems, the performance of a cognitive system must be evaluated on how it decides which action is *appropriate* to take, and not on the merit of *right* or *wrong* choices. The decision made by CEA involves a certain degree of uncertainty that it is, in fact, the best choice at that time. The probabilistic model that is used to make these decisions is influenced by the excitement associated with these choices, which is in turn derived by the Affect Agent. The candidate criteria used to evaluate system performance are:

1. The degree that the response exhibited by the computational model was mediated by the level of the Excitement Meter.
2. The degree of influence on task execution by the probabilistic decision-making derived from this computational model.

Prior to conducting the experiment described above, it was necessary to define initial weightings (A, B) for the various sets of stimuli that can be detected. The Excitement Meter was designed so that it was initially more excited by “fun” tasks such as *playing with toys*, or *listening to music* over performing work (i.e. *executing commands*). The value of A for *playing with toys* was set to twice the value of A for *executing commands*. However, B for *executing commands* was set to half the value of B for *playing with toys*. Using this design, when the experiment was initially run, ISAC chose to play with the toys rather than performing the command. Additionally, the excitement discrepancy initially encoded between the two tasks caused the probability that *playing with toys* was chosen to increase to 100%, and subsequently *executing commands* was decreased to 0%. Figure 16 shows the level of excitement associated with each set of stimuli during the experiment.

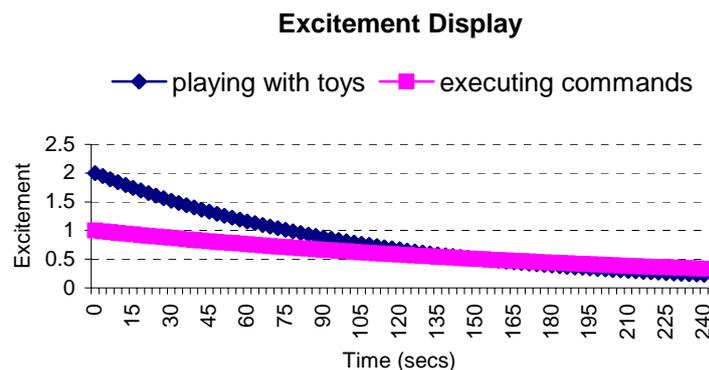


Figure 16. Excitement Levels For Competing Tasks

However, when this experiment was allowed to run for longer than ~3 minutes ISAC’s excitement level dropped sufficiently enough that it then became desirable to pay attention to the person and execute the command. Much of this behavior should be attributed to the fact that the current excitement associations are still heavily dependent on the initial weightings. Multiple trials over multiple days under varieties of circumstances have not yet been performed, therefore, the initial weights of (A, B) have not been modified by the system to reflect ISAC’s personal experiences.

Performance criterion 1 was evaluated based on the effectiveness of equation (2). For this experiment, the computational model worked as intended in representing affect. The fact that certain excitement associations initialize higher or lower and decay faster or slower is the reason why equation (2) was chosen. Since the parameters (A, B) can be retained and updated like standard network weights, this equation adds dynamic flexibility to this system by allowing it to learn excitement associations over time. This can be done by rewarding particular choices and punishing others. Therefore, over time the stimulus *executing commands* may begin to initialize higher than *playing with toys* based on ISAC's own experience.

Performance criterion 2 was evaluated based on the decisions made by the CEA. Using the weights (A, B) in equation (2) the CEA could choose to ignore or accept the suggestions made by the Affect Agent. The values (A, B) represented the strength of the input from the Affect Agent. When the CEA ignores the suggestions it does not allow the probabilistic model to be updated by the Affect Agent. In order to evaluate the performance criterion, in this experiment the CEA was forced to accept the suggestions made by the Affect Agent. Therefore, when both sets of stimuli were present the suggestions passed to the CEA by the Affect Agent caused the probability that *playing with toys* would be chosen to increase to 100% and, conversely, *executing the command* to decrease to 0%. However, due to this drastic change in the action probabilities, the initial decision making within the CEA required no decision. However, later in the experiment a decision was necessary, whether to continue to *play with the toys* or switch to the more exciting *execute the command* task. Figure 16 shows that when this decision was necessary, the associated excitement levels with each task were approximately equal. Likewise, the CEA chose to switch tasks, at this point in the experiment, 50% of the time.

It is important to note that, for completeness, a third choice to *do nothing* should have been made available to ISAC. Future experiments will include this option. Also important in future work is the incorporation of other affect variables, and their role in influencing each other. For example, *fear* of negative reward for not performing a task could negatively influence excitement. *Joy* for being successfully able to *play with the toys* could positively influence excitement, possibly overriding *fear*. In addition, these new variables should also influence the probability that an action is selected. Future work must incorporate updating the parameters (A, B) over longer periods of time, possibly several days. These experiences, over time, are the keys to creating a dynamic affectual system that is individual to ISAC and, to a certain extent, unpredictable by humans.

## 6. FUTURE PLANS

In order to execute a task more robustly, CEA must monitor and evaluate the expected outcome of task execution according to the behavior-percept combination and interrupt task execution if necessary in real. We are currently looking into one real-time Windows operating system offered by German company KUKA. (<http://www.kuka-controls.com>) In addition, ISAC requires past experience as an episode to be used in cognitive control. However, the current representation of episode is too simplistic. We plan to investigate a more robust episodic memory representation and retrieval. Another part of our future work includes expanding the capability of the system to handle more complex tasks that involve multiple percepts and behaviors.

Future work with the self-motivated, internal state-based action selection requires several key components. First, more internal state variables need to be added to the affect agent. Such variables as *fear*, *happiness*, *pain*, etc. would better enable ISAC to make decisions in more complex situations. How to incorporate the variables into the probabilistic decision making model CEA currently uses and how to incorporate the effect these variables have on each other are two important issues to be looked at. Second, when to update the values (A, B) in the excitement equation (2) to reflect a change in ISAC preference is an important issue with the Excitement Meter. These variables represent a key feature of the cognitive system, i.e. the

ability to modify its own preferences based on experience. Third, the question “When should or should not CEA accept input from the Affect Agent?” (i.e. the change in the probabilistic model” needs to be examined closer). Currently, the decision is based on the strength of the excitement association. A more robust solution integrating past experience, knowledge of the current situation, and other affect variables is required. Lastly, the hybrid system nature of the Affect Agent needs further investigation. Currently, events were sets of stimuli defined *a priori*. The real world, however, does not conform to a predefined structure and it is important to understand how to organize low-level events (such as the appearance of a percept) into higher-level events that trigger complex cognitive control mechanisms.

## 7. CONCLUSIONS

During the past decade, we have seen major advances in the integration of sensor technologies, artificial intelligence, and machine learning into a variety of system design and operation. A next challenge will be the integration of human-like cognitive control into system design and operation.

This paper described our efforts to develop the next generation of robots with robust sensorimotor intelligence using a multiagent-based cognitive control. Experiments conducted so far validated the effectiveness of our design.

## ACKNOWLEDGEMENTS

Special thanks to Flo Wahidi, and the graduate students at the Center for Intelligent Systems. This work is supported in part under NSF grant EIA0325641, “ITR: A Biologically Inspired Adaptive Working Memory System for Efficient Robot Control and Learning”.

## REFERENCES

- [1] M.S. Gazzaniga, Cognitive Neuroscience. The Biology of the Mind, 2<sup>nd</sup> ed., Norton & Co., New York, 2002, p. 615.
- [2] J.R. Stroop, “Studies of interference in serial verbal reactions,” *Experimental Psychology*, 18, 1935, pp. 643-661.
- [3] M.M. Botvinick, T.S. Braver, D.M. Barch, C.S. Carter, J.D. Cohen, “Conflict monitoring and cognitive control,” *Psychological Review*, 108(3), 2001, pp. 624-652.
- [4] K. Kawamura, R.A. Peters II, R. Bodenheimer, N. Sarkar, J. Park, A. Spratley, K. A. Hambuchen, “Multiagent-based cognitive robot architecture and its realization,” *Int’l Jo. of Humanoid Robotics*, 1(1), 2004, pp. 65-93.
- [5] J.M. Mendel and K.S. Fu, eds., Adaptive, Learning and Pattern Recognition Systems: Theory and applications, Academic Press, New York, 1970.
- [6] E.K. Miller, C.A. Erickson, R. Desimone, “Neural mechanisms of visual working memory in prefrontal cortex of the macaque,” *J. Neurosci.* 16 (16), 1996, pp. 5154-5167.
- [7] T. Pack, D.M. Wilkes, K. Kawamura, “A software architecture for integrated service robot development,” *Proc. of IEEE Trans. on Systems, Man and Cybernetics*, Oct. 1997, pp. 3774-3779.
- [8] J.S. Albus, “Outline for a theory of intelligence,” *Proc. of IEEE Trans. on Systems, Man, and Cybernetics*, 21(3), 1991, pp.473-509.
- [9] R.A. Peters II, K.A. Hambuchen, K. Kawamura, D.M. Wilkes, “The sensory egosphere as a short-term memory for humanoids,” *Proc. of the IEEE-RAS Int’l Conf. on Humanoid Robots*, Waseda University, Tokyo, Nov.22-24, 2001, pp.451-459.
- [10] A.D. Baddeley, Working Memory, Oxford: Clarendon Press, 1986.

- [11] K.A. Hambuchen, "Multi-modal attention and binding using a sensory egosphere", Ph.D. Dissertation, Vanderbilt, Nashville, TN, May 2004.
- [12] J.L. Phillips, D. Noelle, "A biologically inspired working memory framework for robots", *Proc. of the 27<sup>th</sup> Annual Conf. of the Cognitive Science Society*, 2005, pp. 1750-1755.
- [13] S.M. Gordon, J. Hall, (2006), "System integration with working memory management for robotic behavior learning", *Proc. of the 5<sup>th</sup> International Conference on Development and Learning*, Bloomington, IN, June 2006.
- [14] C. Rose, M.F. Cohen, B. Bodenheimer, "Verbs and Adverbs: Multidimensional motion interpolation," *IEEE Computer Graphics and Applications*, vol. 18, no. 5, September–October, 1998, pp. 32-40.
- [15] M.P. Shanahan, "A cognitive architecture that combines internal simulation with a global workspace", *Consciousness and Cognition*, vol. 15, 2006, pp. 433-449.
- [16] M.A. Arbib, "Perceptual Structures and Distributed Motor Control," in *Handbook of Physiology-The Nervous System II: Motor Control*, ed. V. B. Brooks, American Physiological Society, Bethesda, MD, 1981, pp. 1449-80.
- [17] R.A. Brooks, "A robust layered control system for a mobile robot," *IEEE journal of Robotics and Automation*, Vol.RA-2, No. 1, 1986, pp. 14-23.
- [18] R.W. Picard, *Affective Computing*, MIT Press, Cambridge, MA. 1997
- [19] P. Ratanaswasd, C. Garber, A. Lauf, "Situation-Based Stimuli Response in a Humanoid Robot", *Proc. from 5th Int'l Conf on Development and Learning*, Bloomington, IN, June 2006.
- [20] J.R. Anderson, D. Bothell, M. Byrne, S. Douglas, C. Lebiere, Y. Qin, "An integrated theory of the mind", (ACT-R), *Psychological Review*, 111(4), 2004, pp. 1036-1060.
- [21] M. Branicky, "Analyzing continuous switching systems: Theory and examples", *Proc. of American Control Conference*, Baltimore, MD, June 1994
- [22] A. Cohen, R. Shoup, "Response selection processes for conjunctive targets", *Jo. of Experimental Psychology: Human Perception and Performance*, vol. 26, 2002, pp. 391-411.
- [23] A.W. Spratley II, "Verbs and adverbs as the basis for motion generation in humanoid robots", M.S. Thesis, Vanderbilt University, Nashville, TN, August 2006.
- [24] O.C. Jenkins, M.J. Matarić, "Automated derivation of behavior vocabularies for autonomous humanoid motion," *2nd Int'l Joint Conf. on Autonomous Agents and Multiagent Systems*, Melbourne, Australia, July 14-18, 2003.

## APPENDICES

### APPENDIX 1. Spatial Attention and Action Selection

Humans pay attention by emphasizing the locations of percepts with high saliency. This process is known as *spatial attention* [22]. In our architecture, spatial attention is realized through assigned *Focus of Attention* (FOA) by the Attention Network [11]. The percepts in attention then are brought into working memory for further processing. During an event, past episodes are retrieved from Episodic Memory using cues such as percepts and task information. Given the event and current state, the system will need to make decision using the action performed during these past episodes. The actions will be extracted from episodes, sorted, and given priority based on cues such as affective values and rewards of the episode from which they are extracted.

The action selection process retrieves episodes using cues such as the affect value. Retrieved episodes then are assigned probabilities as follows: Let  $p_j$  be the priority for the  $j$ -th action and  $p_T = \sum_{j=1}^N p_j$ , where  $N$  is the number of the retrieved actions, then action  $A_j$  will be assigned the probability

$$P[A_j] = \frac{p_j}{p_T}; j=1,2,\dots,N, \quad \text{where} \quad \sum_{j=1}^N P[A_j] = 1. \quad (5)$$

The action selection process then will be performed probabilistically as follows: The unit interval  $[0,1]$ , representing the summation of  $P[A_j]$ , is partitioned into  $N$  regions, and the  $j$ -th region has a width of  $P[A_j]$ . A uniform random number  $R$  is generated,  $0 \leq R \leq 1$ . Let  $T(0) = 0$ . For each region  $j$ , compute the boundary of  $P[A_j]$  as  $T(j-1)$  to  $T(j)$ , where

$$T(j) = \sum_{k=1}^j P[A_k]. \quad (6)$$

If  $T(j-1) \leq R \leq T(j)$ , select the  $j$ -th action. Figure 17 illustrates the action selection process currently used.

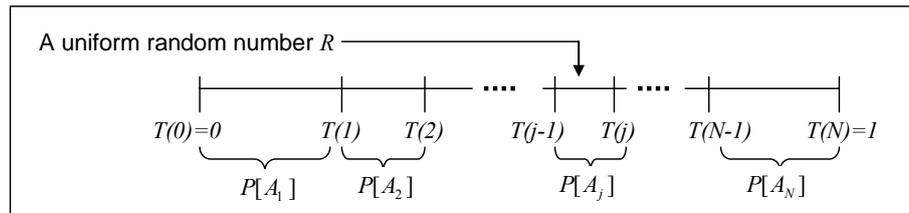


Figure 17. Probabilistic Action Selection Process

## APPENDIX 2. Verbs and Adverbs for Behavior Execution

The Verbs and Adverbs algorithm is a motion interpolation technique originally developed for computer graphics by Rose, et al [14]. In this technique motion exemplars are used to construct *verbs* that can be interpolated across different spaces of the motion represented by the *adverbs*. An important aspect in storing and re-using a motion for a *verb* is the identification of the *keytimes* [23] [14] of the motion. The keytimes represent significant structural breaks in the particular motion. For the Verbs and Adverbs technique to function properly individual motions for the same verb must have the same number of keytimes and each keytime must have the same significance across each motion. Figure 18 shows keytimes for three example motions. The example motions are recording of the same motion, three different times. This information is used to create the verb, *handshake*. The keytimes in this example are derived by analyzing the motions using a technique called Kinematic Centroid [24]. The x-axis represents the normalized point index for each motion. The y-axis represents the Euclidian distance of the kinematic centroid of the arm from the base of the arm.

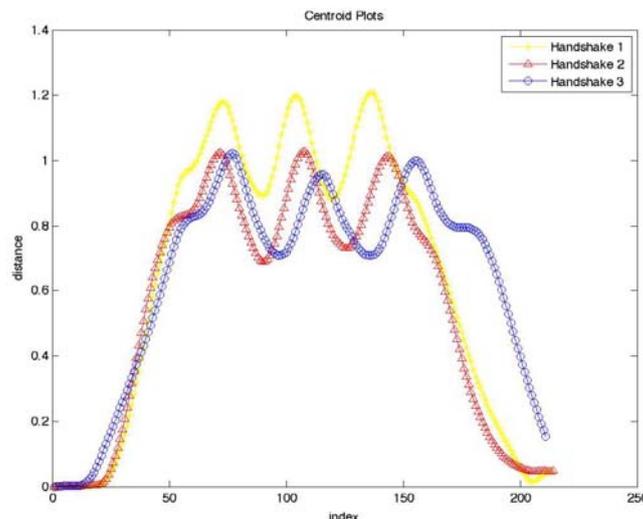


Figure 18. Example Motions and Keytimes [23]

Each verb can have any number of adverbs, each of which relate to a particular space of the motion. For example, the verb *reach* could have two adverbs: the first related to the direction of the *reach* and the second related to the distance from ISAC's origin that the particular motion is to extend. Extending this example, adverbs could be added to include features from any other conceivable space of the motion, such as the strength of the motion or the speed of the motion. Stored in the LTM are the verb exemplars and the adverb parameters for each verb. New motions such as *reaching*, or *handshaking* are interpolated by ISAC at run time using the new (desired) adverb values. One important point, in our system new motions are never extrapolated. This is due to the fact that extrapolated motions can potentially lead to undesirable (or unachievable) arm configurations. Currently, ISAC is using the Verbs and Adverbs algorithm for three behaviors: *reach*, *handshake*, and *wave*.

### APPENDIX 3. Memory Contents During WM Training

Table 2 shows the contents of short-term and long-term memory during the experiment discussed in Section 3. During the experiment, two bean bags were present in front of ISAC. Additionally, three behaviors had been trained and placed in long-term memory. This information was encoded into working memory "chunks", void data structures. The WM then chose from these chunks and the contents of WM guided task execution. In other words, if the chunks *reach* and *blue bean bag* were present then ISAC reached to the blue bean bag. Table 3 shows example contents of working memory during four of the training trials. When one percept and one behavior chunk was not present, the missing chunk(s) were filled in at random.

SES	LTM
1. bean bag: location = (Figure 6.b), type = Blue	1. <i>reach</i>
2. bean bag: location = (Figure 6.a), type = Red	2. <i>handshake</i>
	3. <i>wave</i>

Table 2. Memory Contents During Simulation Training

Trial #:	Working Memory Contents			
	1	2	3	4
Chunk 1	Blue bean bag	Red bean bag	<i>Wave</i>	<i>Handshake</i>
Chunk 2	<i>reach</i>	Blue bean bag	Blue bean bag	Red bean bag
Random:	NA	<i>Wave</i>	NA	NA
Reward:	20	0	0	0

Table 3. Working Memory Contents During Simulation Training

### APPENDIX 4. Perception Encoding Used in FRA Experiment

#### 4.1 Sound Localization

Within the ISAC cognitive system, the location of a loud noise, such as hand clapping, is used as a means of focusing attention and a cause for invoking the reactive response in FRA. The location of the sound source is detected using sound localization. The basic configuration

of the sound localization system used includes a pair of microphones, located a finite distance apart as illustrated below. The sound waves arrive at the microphones at different times. Using this time difference, the probability of the clapping sound found at any given location in the environment is calculated and the result is sent to the SES.

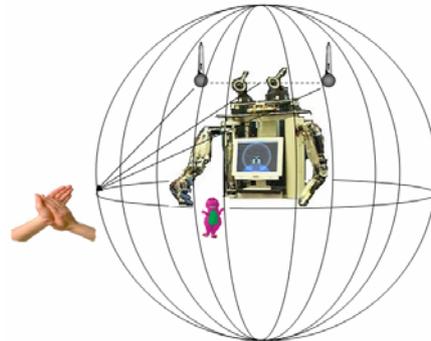


Figure 19. Sound Localization

#### 4.2 Motion Detection

Motion detection is performed by the motion detection percept agent that utilizes a laser range finder (LRF) mounted above the cameras on the head. The laser range finder is positioned so planar scans of the area in front of the robot can be obtained from various angles. Successive scans from the same angle are subtracted to obtain a temporal difference map. The positions of non-zero values along the temporal difference map, the angle of the planar scan, and the range data are used to indicate the position of motion in front of ISAC which the information is posted to the SES.

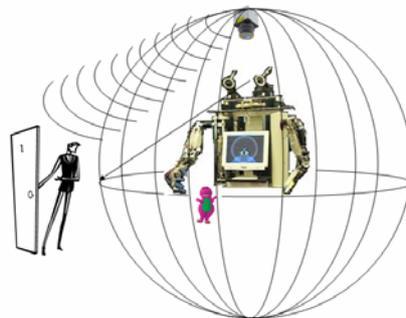


Figure 20. Motion Detection