Implementation of Cognitive Control for a Humanoid Robot

Kazuhiko Kawamura, Stephen Gordon, Palis Ratanaswasd, Erdem Erdemir, and Joe Hall
Center for Intelligent Systems
Vanderbilt University
Nashville TN 37235-0131 USA

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. We are interested in implementing such cognitive control functionality in robots. This paper outlines the multi-agent-based, hybrid cognitive architecture for a humanoid robot and the progress made on the implementation of cognitive control functionalities using attention, affect, working memory and internal rehearsal.

KEYWORDS: humanoid robot, cognitive control, working memory, episodic memory, affect, task switching, self-motivation, internal rehearsal

1. INTRODUCTION

As the need to control complex robotic systems increases, it is important to look beyond engineering- and computer science-based approaches. For example, humans have the capacity to receive and process enormous amount of sensory information from the environment, exhibiting integrated complex sensory-motor associations as early as two years old [Gazzaniga, 2002]. 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 task learning and execution. Specifically, we see cognitive flexibility and adaptability in the brain as desirable design goals for the next generation of intelligent robots.

In 2004, we introduced a multi-agent-based, hybrid robot control architecture with memory structures [Kawamura, 2004]. In this paper, we will present the progress made since then on the architecture, cognitive control, sensory-motor binding, task execution and switching, internal rehearsal, and a self-motivated action selection mechanism.

2. COGNITIVE CONTROL FOR ROBOTS

Engineers have long used control systems utilizing feedback loops to control mechanical systems. Limitations of model-based control led to a generation of *intelligent control techniques* such as fuzzy control, neuron-computing and reconfigurable control [IEEE Control Society]. The human brain, on the other hand, 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

[Botvinick, 2001]. We are interested in realizing this cognitive control capability in our humanoid robot. Figure 1 illustrates a conceptual model of cognitive control process, which we are using to realize cognitive control for ISAC humanoid robot.

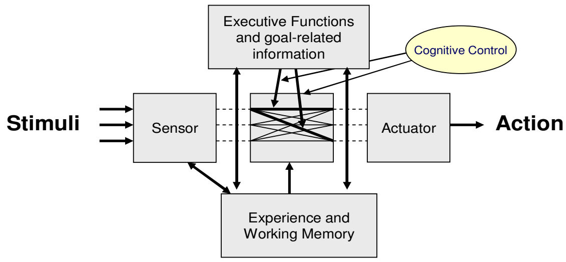


Figure 1. Model of Cognitive Control. Modified from Miller, et al [Miller, 2003]

As the complexity of tasks grows, so does the software complexity necessary to provide robust sensory-motor coordination. During earlier development of our humanoid robot, it was realized that development and maintenance of complex software systems could benefit from domain-specific guidelines that promote code reuse and integration through software agents. This led us to develop a multiagent-based robot control architecture based on the Intelligent Machine Architecture (IMA) [Pack, 1997]. IMA is designed to provide software platform that allows anyone to developed own structure using atomic agents. IMA (website: http://eecs.vuse.vanderbilt.edu/cis/concepts/ima.shtml) allows for modular design and the development of subsystems from perception modeling to behavior control through the collections of software agents and associated memories. Figure 2 was configured using IMA agents and associated memory structures.

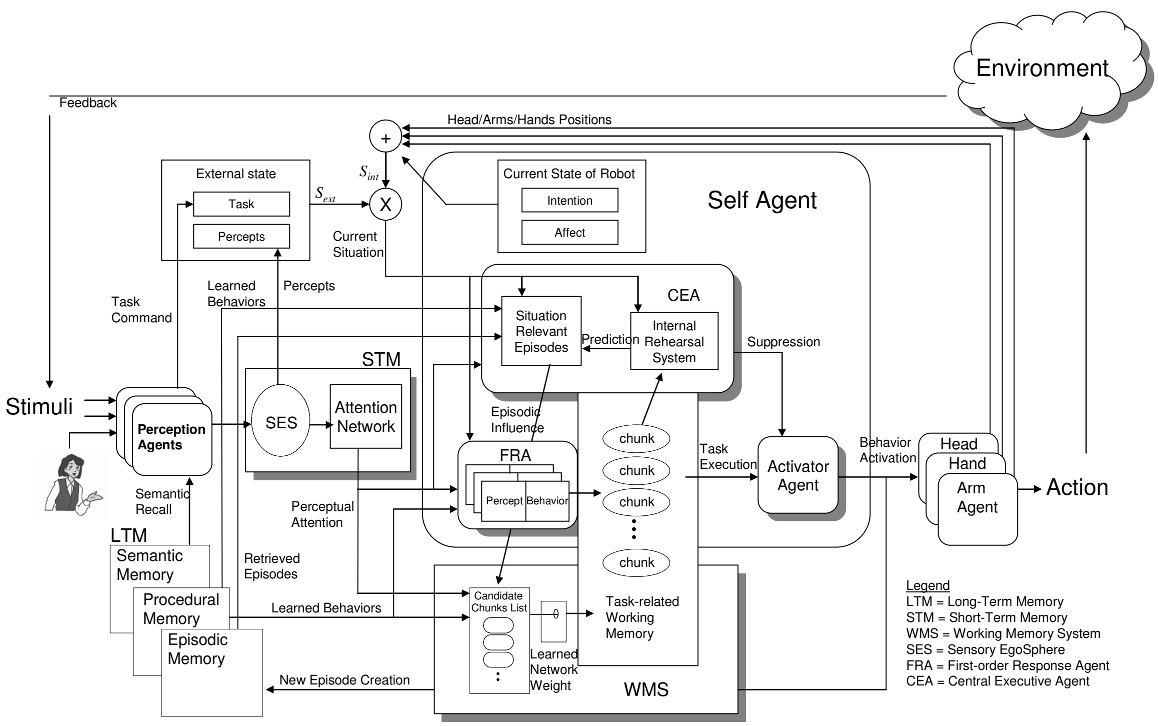


Figure 2. Multiagent-Based Cognitive Robot Architecture

For any learning system, memory plays an important role. As Gazzaniga, 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." [Gazzaniga, 2002, p. 302]. ISAC memory structure is divided into three: 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) and serves as a spatial-temporal STM [Peters, 2001]. LTM stores information such as *skills learned* and *experiences gained* for future recall. WMS is a goal-directed subsystem that performs task-critical operations on the contents of working memory chunks. Operations are executed using the Working Memory Toolkit [Phillips and Noelle, 2005].

In our architecture, cognitive control is implemented using agents such as the Central Executive Agent and the First-Order Response Agent, and other modules such as the Attention Network, WMS and the Episodic Memory as discussed in later sections.

3. WORKING MEMORY SYSTEM

There is extensive evidence that the brain contains a memory system that actively maintains a small amount of task-essential information in a manner that allows that information to guide attention, focus learning efforts, and generally support the execution of tasks [Waugh and Norman, 1965]. Driven by this evidence, we have constructed a robotic working memory system whose components are fabricated by a software toolkit [Phillips and Noelle, 2005]. This section describes how this neuroscience-inspired working memory toolkit has been used to train percept-behavior binding within the cognitive architecture.

3.1 Pre-Frontal Cortex Working Memory Model

Working memory (WM) may be considered as a short-term memory cache that actively maintains information relevant to the current task for a short period of time. There is evidence from neuroscience that WM is distinct from other forms of memory, and that the prefrontal cortex (PFC) plays an important role in WM [O'Reilly, et al., 1999]. More recently, Kobayashi demonstrated that dorsolateral prefrontal cortex (DLPFC) is responsible for task execution in organizing human behaviors spatially and temporally by measuring the level of oxy-hemoglobin in DLPFC during cognitive task execution [Kobayashi, 2007].

The function of WM is based on the expectation of future reward. There is reason to believe that the neurotransmitter dopamine plays an important role in this reward-based learning [McDonald, et al., 2000]. Inspired by this, a PFC-based computational working memory model based on a neural network and temporal difference (TD) learning [Sutton, 1988] has been developed at Vanderbilt [Phillips and Noelle, 2005].

Another important concept of WM is its size (or capacity). Early research on working memory suggested the capacity as "seven plus-or-minus two" [Miller, 1956], however recent research suggests this number may be closer to four [Cowan, 2001].

3.2 Working Memory System Training for Percept-Behavior Association Learning

One of the best known working memory models is the behavioral study-based framework of Baddeley and Hitch, in which a central executive controls two separate subsystems [Baddeley, 1986]. More recently a third structure, an episodic buffer, has been added to the framework [Baddeley, 1990]. Working memory system within ISAC was inspired by their work and is implemented using the Central Executive Agent, the Attention Network and the Working Memory Toolkit.

3.2.1 Percept-Behavior Association Learning

The Attention Network [Hambuchen, 2004] assigns the *focus of attention* (FOA) to recently processed percepts as described in Section 4. FOA-indexed percepts are then sent to the working memory as the candidate *chunks for percept-behavior association learning* as shown in Figure 3. For the remainder of this paper, the term b*ehavior* denotes the stored information in procedural LTM. This behavior information represents the raw data used to generate a motion along with a unique identifier. S*kill,* on the other hand*,* relates to the ability of the cognitive processes to use these stored behaviors to accomplish tasks.

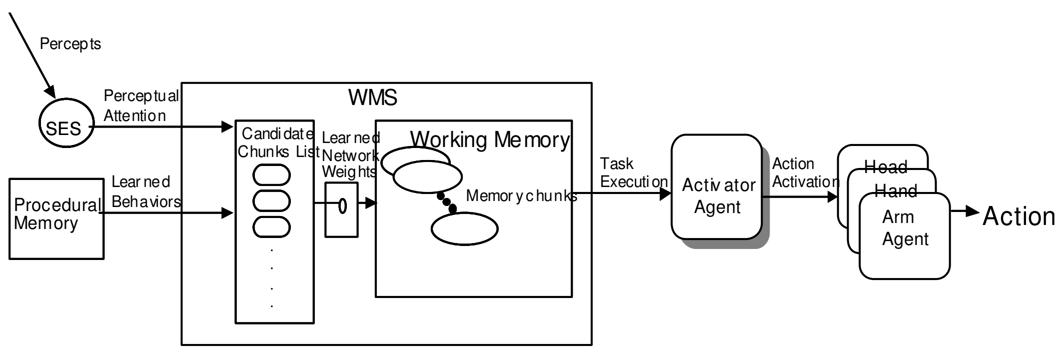


Figure 3. Percept-Behavior Association Learning Loop

Working Memory System (WMS) training was done using the Working Memory toolkit (WMtk). This toolkit is based on the computational model of working memory [Braver, et al., 2000] that attempts to load chunks of information from a candidate chunks list (Appendix 2). The size of WM, i.e. the number of chunks, is dependent on the type of task being attempted. For the experiment described in this section the size was two. WMS also aids the development of symbolic grounding [Harnad, 1990] in our system. For example, when ISAC receives a command *reach to bean bag,* WMS associates this command with one (or more) of the behaviors stored in the Procedural Memory. When ISAC operates on the symbol *reach* there is a direct link with the raw data, or trajectory information, used to move the arm. If this link, i.e. association, helps task execution, then WMS maintains the chunks for the duration of the task. Finally, WMS facilitates the development of association connections between different types of chunks for a particular task. This final point will be discussed further in Section 3.3.

3.2.1 Task Category

WMS faces many of the same computational constraints that other machine learning algorithms face. Primarily, it is intractable to encode all possible states or situations that ISAC may encounter within a single instance of WMtk. Therefore tasks have been grouped into a set of categories and single instances of the WMtk were used for similar categories of tasks. For example, the tasks *reach to bean bag* and *play with LEGO toy* both involve one object and one skill, and thus can be categorized together as one category. A more complex task requires more than one object or one skill and therefore would require a separate instance of WMtk. The goal of separating working memory in this manner is to avoid the time delays associated with the potentially exponentially growing state and chunk comparison. For our work, we are only dealing with tasks belonging to one of the following four categories to reflect the types of tasks which ISAC generally encounters: Category1 tasks are a combination of a single object and a single skill. Category2 tasks represent the Category2 tasks represent the situation where ISAC is performing simultaneous is performing simultaneous single object - single skill tasks. For example, when ISAC uses both arms to manipulate one object, the task is considered to be Category2. Category3 and Category4 represent situations in which multiple skills are necessary or multiple objects are present. In the future, more complex tasks will be considered using semantic net and schema representations as shown in Figure 4.

Task category:

* Category1 : {object} x {skill}
* Category2 : [{object} x {skill}] x [{object} x {skill}]
* Category3 : {one object} x {multiple skills} or {multiple objects} x {one skill}
* Category4 : {multiple objects} x {multiple skills}

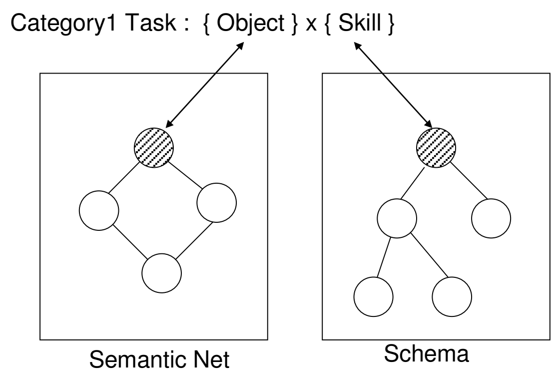


Figure 4. Task Representation using semantic net [Shapiro, 1971] and schema [Platt, et al., 2006]

The object-skill association training in this section represents ISAC's initial trial-and-error learning to a new Category1 task. For each experiment, WMS initializes an untrained instance of working memory. New task is defined to be a task for which no similar episode is found in the Episodic Memory. After a behavior has been selected to represent a skill, the motion interpolation and execution of the behavior to act on the object in the environment was performed using a modification of the Verbs and Adverbs algorithm [Rose, et al., 1998] discussed in Appendix 2. An example of interpolated behavior is shown in Figure 5.

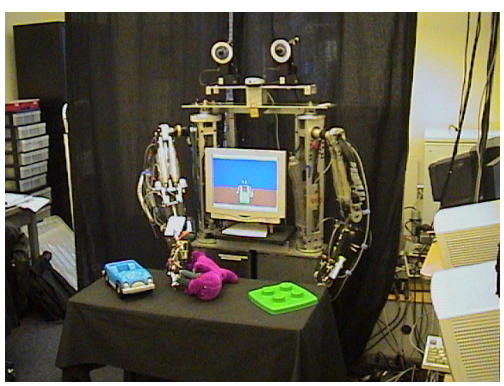


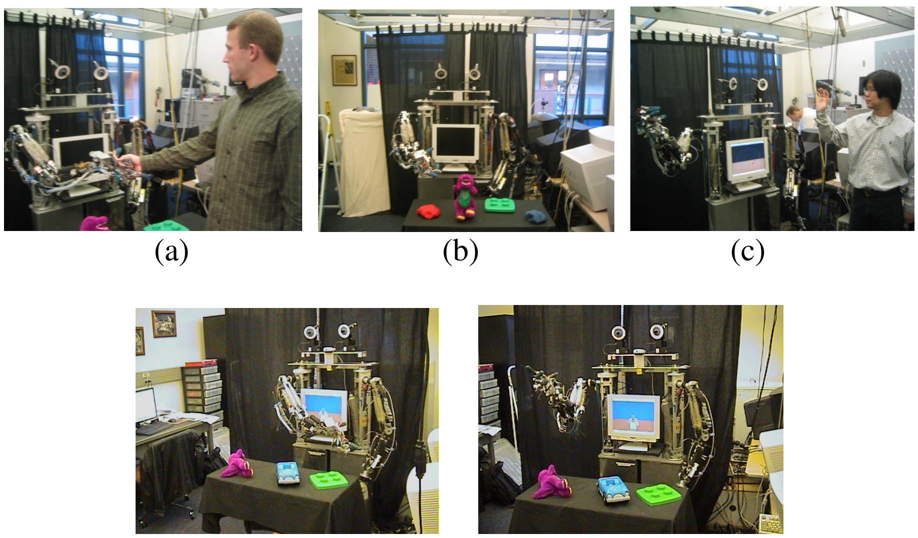
Figure 5. Interpolated *Reach to Barney* behavior

3.2.2 Experiment

In this experiment, WM was trained to perform the task to *reach to bean bag* (a Category1 or a Category3 task) from the home position, i.e. arms in stationary position. Also ISAC is assumed to be right-handed, i.e. ISAC will use the right arm. In the trials, two bean bags were placed on a table in front of ISAC and WMS was required to choose one. Preliminary results for training WM have been presented in [Gordon and Hall, 2006].

First, several behaviors were taught to ISAC through teleoperation and encoded within the Procedural Memory. Each behavior was numbered in the order in which that behavior was trained, e.g., the first behavior trained was given the name "Behavior1". For easier understanding, these behaviors are listed below using a symbolic name and shown in Figure 6:

Behavior1 -Handshake
Behavior2 -General (three dimensional) reaching
Behavior3 -Wave
Behavior4 -Reach left and right on a table
Behavior5 -Reach up and down



(a) (b) (c)

(d) (e)

Figure 6. Sample Configurations for Behaviors used: (a) Behavior1, (b) Behavior2, (c) Behavior3, (d) Behavior4, (e) Behavior5

Since WMtk uses a TD learning algorithm, it is necessary to provide a task-specific reward rule. This reward rule was embedded within WMS that rewarded each chunk selection based on the success of the current trial. The specific reward used is discussed in Section 3.3. WMtk also allows the use of an exploration percentage. This percentage specifies how often WMS should choose random chunks rather than the chunks for which it expects a reward. This is intended to avoid local maxima by encouraging random exploration even after learning has been accomplished [Phillips and Noelle, 2005]. The exploration percentage was set to 15% during the experiment.

3.3 System Performance

The performance of the working memory system was evaluated using the following criteria:

1. The ability to choose the appropriate behavior chunk to accomplish the task.
2. The ability to choose the appropriate percept chunk to accomplish the task.
3. The ability to use a higher decision rule to distinguish among similar performances (e.g. reaching to the nearest bean bag rather than the farthest one).

For the given task to *reach to bean bag*, one behavior from the set in Figure 6 must be chosen to associate with the "reach" skill along with one percept from SES. For each candidate percept-behavior combination, the following reward wad given:

$$R = \frac{b}{c + Dist} \qquad (1)$$

where *Dist* is the minimum Cartesian distance from the hand to the bean bag during task execution. The terms *b* and *c* are constants that define the maximum reward possible. This reward rule is to let WMS distinguish between two seemingly correct percept-behavior associations in order to find the best action (Gordon and Hall, 2006). For example, if two bean bags were present, i.e. a Category3 task, this reward rule should guide WMS to choose to reach to the *nearest* bean bag.

Initial trials for this experiment were performed in simulation in order to speed-up the testing phase of learning. Percept identification, selection, and tracking were all performed by the system at run-time, but interpolated actions were not performed assuming that the actuators and low-level arm controllers performed correctly. Appendix 3 shows sample contents of SES, LTM and WM during this experiment.

Figure 7 shows the results for learning the correct behavior for the given task, *reach to bean bag*. For this experiment, the bean bag was randomly placed in front of ISAC. After several exploration, the system converged on Behavior2 (three-dimensional reach motion) except for occasional exploration.

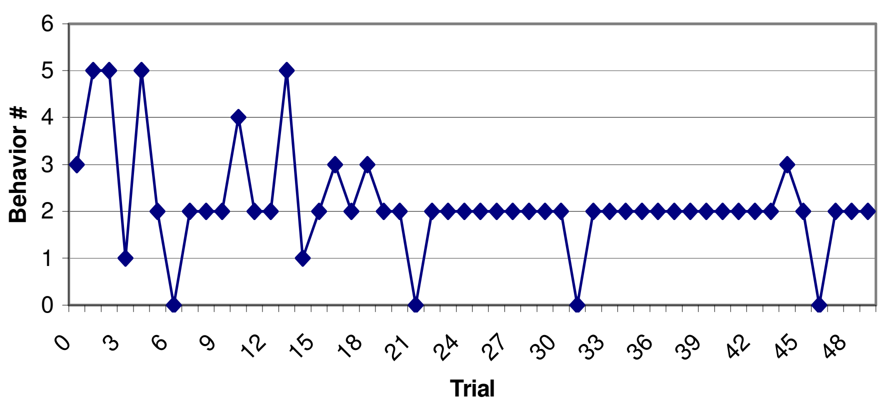


Figure 7. Behavior Selection for the *Reach to Bean Bag* Task.

This system is also capable of distinguishing between similar percept-behavior learning. Figure 8 shows a different training curve for learning to *choose the nearest bean bag*. This was performed after the correct behavior had been learned. In this figure, Percept1 is the nearest bean bag and Percept2 is the bean bag that is farther away. The system started by randomly choosing percepts and tried to reach. However, after further trials the system converged on the correct bean bag. Finally, it is important to note that during exploration WMtk did consider the option of not loading any chunks (i.e. Percept0 in Figure 8) a possibility. During these trials no reward was received.

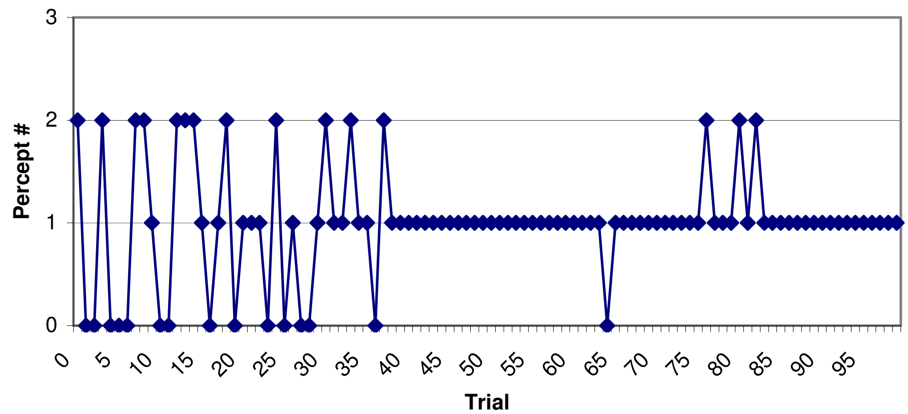


Figure 8. Learning to Choose *the Nearest Bean Bag*.

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

A cognitive robot should be able to make decision and act accordingly based on the situation it is in and its internal states. The representation of internal states within the ISAC cognitive architecture is an IMA agent called the Self Agent. The Self Agent is a virtual agent consisting of a number of atomic agents that maintain tight communications among them in order to share and act on a common set of information. Among them, the Central Executive Agent (CEA) and the First-order Response Agent (FRA) play important roles on decision making and task execution. Their roles are described below.

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

Figure 9 shows the current structure of the Self Agent. The Central Executive Agent (CEA) is responsible for cognitive control during task execution. It makes decisions and invokes skills necessary to perform the given task using the Focus of Attention (FOA) and past experience. CEA operates in accordance to the *intention* which the Intention Agent interprets from the task command. Decision making within CEA is mediated by *affect* which is managed by the Affect Agent. The Activator Agent invokes head, arm and hand agents to generate actions.

The First-order Response Agent (FRA) is responsible for generating both routine and reactive responses. The term *first-order response* was originally used by Shanahan [Shanahan, 2006] for reactive responses as opposed to the higher-order cognitive response. (Section 5 provides further discussion on this.) In our implementation, FRA handles reactive responses by invoking corresponding behaviors when certain percepts receive the focus of attention. Reactive responses within FRA is implemented as a multithreaded process where the associations between percept-behavior are embedded within separate running threads. Salient percepts on SES are put in the Focus of Attention (FOA) by the Attention Network. Each running thread compares the most salient percept from the candidate percept-skill pairs. If a matching if found, FRA posts both the percept and the skill onto the working memory (WM) as chunks. The Activator Agent then takes the chunks from WM and distributes them to relevant atomic agents for action.

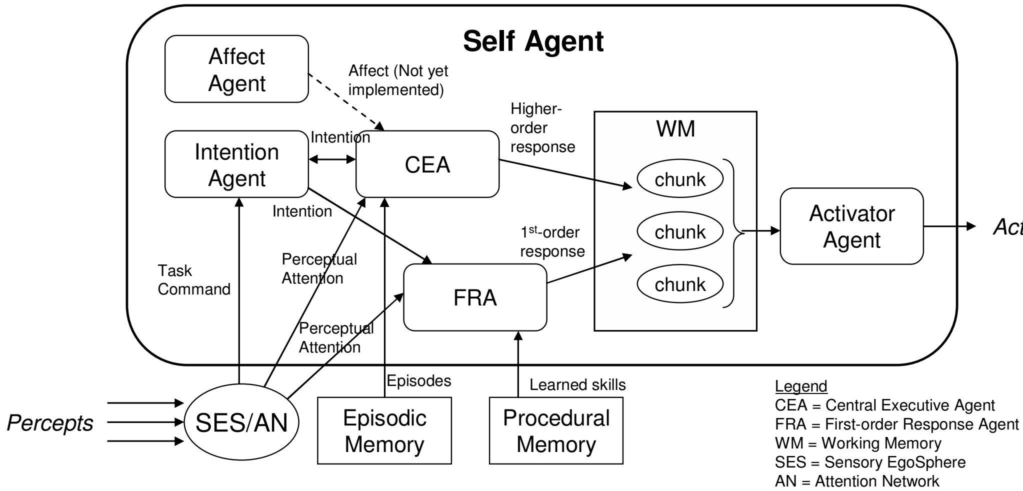


Figure 9. Structure of the Self Agent

Besides reactive responses, FRA maintains one thread that is responsible for routine task execution. This thread invokes corresponding skills when the current task command matches the learned task in LTM. Note that the current task could be assigned externally by a human or internally generated by self-motivation as discussed in Section 6. FRA then posts the behavior found in the learned task and the percept in FOA into WM as chunks where the Activator Agent uses the chunks similarly to the case of reactive responses. This routine response thread will be suppressed whenever any one of reactive response threads become active as in the case of Brook's Subsumption Architecture [Brooks, 1986].

If FRA finds no matching skill to the task command, task execution will be handled by the Central Executive Agent which uses the past experience and the current situation as a separate cognitive process. Past episodes that contain the similar task information as the current task will be recalled and the behaviors that are used to perform tasks in these episodes will be considered based on maximum likelihood. The candidate skills and the percept on SES will then be posted on the working memory and the Activator Agent will activate one skill-percept pair based on its maximum likelihood of success.

4.2 Task Switching Experiment

A two-part experiment was conducted to validate how FRA handles the routine and reactive responses. Figure 10 shows the FRA response process involved in the experiment.

4.2.1 Routine-to-Reactive Response Experiment

The first part of the experiment was conducted to validate the cognitive loop to execute a Category1 task using a routine response and the ability to maintain the task context after a reactive response is invoked.

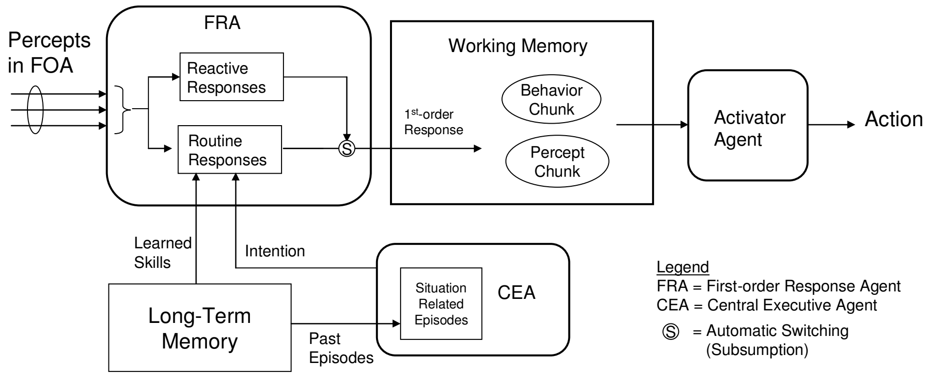


Figure 10. First-order Response Loop

Experimental steps:

1. ISAC actively monitors the environment around the entrance door on the right hand side using its laser scanner.
2. Barney doll is placed within the field of view of head cameras, prompting ISAC to play with the doll according to its innate preference.
3. When someone claps the hands, ISAC react to the noise, stops playing with Barney and saccades toward the source of the sound.
4. Since the current 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 to validate the functionality of FRA to switch tasks when a new situation is recognized.

Experimental steps:

1. ISAC continues the task of playing with Barney.
2. Someone enters the room and approaches ISAC.
3. When a motion is detected, ISAC stops executing the current task, and fixates the cameras on the detected motion.
4. When the moving object, i.e. a person, enters the workspace, ISAC recalls a similar experience with the person and executes the *handshake* skill instead of going back to the previous task.

Figure 10 shows the lab view during the experiment.

(a) (b) (c) (d)

Figure 10: Lab views during experiment - ISAC (a) Plays with Barney using stereo vision , (b) Responds to clapping sound, (c) Detects motion using a laser scanner, (d) Shakes hands with the person

4.3 System Performance

4.3.1 Routine-to-Reactive Response Experiment

This experiment shows how the cognitive cycle to perform task switching involving CEA and FRA works. CEA was used to generate the task command internally and responsible for task switching. The operation was evaluated using the following criteria:

1. The ability to perceive the environment using a variety of sensors..
2. The ability to use routine responses to execute tasks.
3. The ability to switch tasks based on an event or situational change.
4. The ability to switch back and forth between reactive and routine responses.

To take advantage of a multiagent architecture, each stimulus is handled by a different perception agent with its own signal processing techniques embedded. This allows ISAC to expand its perception capability by easily adding more sensors in the future. The outputs from perception agents govern how ISAC perceive its environment. In this experiment, a sound (in addition to human voice) detection agent was newly created to detect the sound of clapping hands. The quality of the clapping sound depends mainly on the distance between the sound source and the microphones, but also it depends on how different people produce hand clapping sounds. Data from the two microphones on ISAC were processed using the energy level, the histogram and the angle of the sound. A set of clapping sounds were given at various angles as shown in Figure 12.

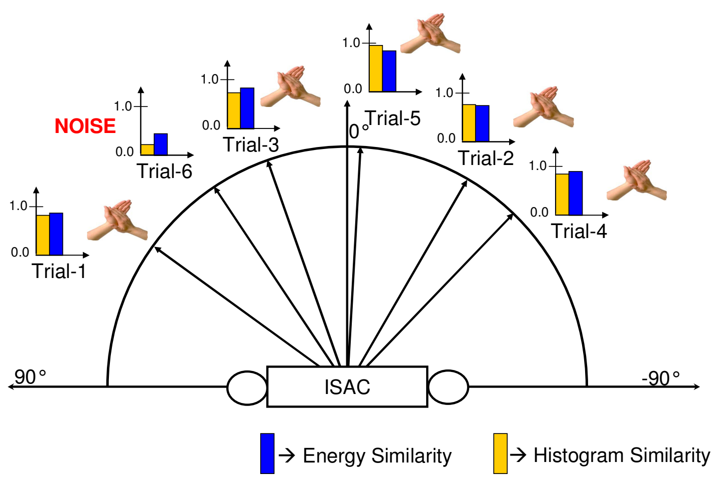


Figure 12. Hand Clapping Experiment

When a clapping sound was detected, it was posted on SES as a percept. FRA reactively responded to the percept by posting the percept and its associated behavior into WM as chunks. FRA removed the chunks when the percept disappeared and ISAC went back to its previous action. 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)	Histogram Similarity (0-100)	Energy Level (0-100)	Detected (yes/no)	Probability Measure	Response Time (ms)	Resuming Time (ms)
1	64.07	83	88	yes	0.84	102	153
2	-38.65	78	75	yes	0.87	106	143
3	18	76	86	yes	0.89	105	135
4	-49	86	93	yes	0.85	114	141
5	-2.65	93	85	yes	0.92	96	139
6	27	23	45	no	0.24	N/A	N/A

Table1. Hand Clapping Experimental Results

A set of built-in percept-behavior associations represented the reactive responses. The behavior is executed as soon as the percept appears on SES, but because of the complexity of the sound detection algorithms and the communication delay time, a small delay (i.e. the time the system took to invoke a reactive behavior and the time it resumed the previous routine behavior) occurred as shown in Table 1.

4.3.2 Task Switching Experiment

The second part of the experiment used motion detection as a cue that alerts the system of other events in the environment that may require attention. Since ISAC is a stationary robot, it is vigilant of people entering and leaving the room. Figure 13 shows several paths of the detected motions. FRA innately reacts to a motion at the area around the door to detect any motion. It is assumed that ISAC has positive experiences to greet people when someone approaches. CEA thus recalls such episodes and greets the when the person enters ISAC workspace.

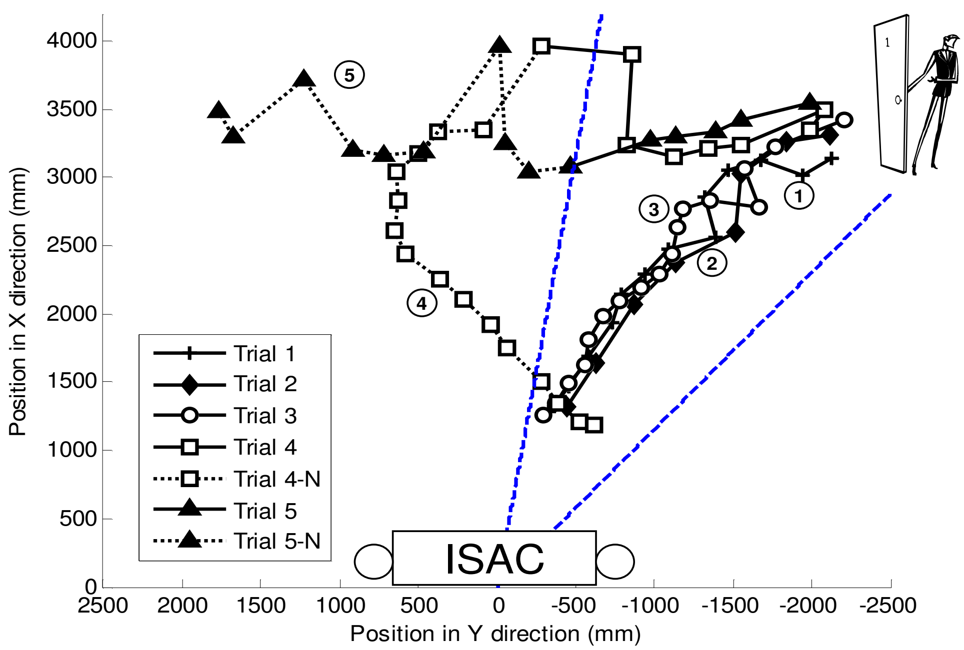


Figure 13. Motion Detection Experimental Results - Trial 1: Approaching with normal speed,

Trail 2: Approaching fast, Trial 3: Approaching slowly, Trial 4: Approaching from the left, and Trial 5: Walking across the room

As shown in Figure 13, ISAC only responded to the motions that stopped within its workspace and approached it from the right. (Remember, ISAC was actively monitoring the right hand side of the environment for any movement.) These motions are indicated by the solid line. The motions that were ignored are shown in dotted lines. When ISAC decided not to greet, it went back to perform the previous task.

5. INTERNAL REHEARSAL USING COLLISION DETECTION

Humans are able to have sensory experiences in the absences of external stimuli [Ziemke, et al, 2002]. This has been illustrated by experimental results of, e.g., Lee and Thompson [1982]. It thus seemed reasonable to assume the existence of an 'inner world' where sensory experiences and consequences of different behaviors may be anticipated. The idea of the existence of such an 'inner world model', however, has been questioned since the mid-1980s by a number of researchers (e.g., Brooks, 1986; Clancy, 1997; Clark, 1997; Pfeifer and Scheier, 1999) who de-emphasize the role of internal world models and instead emphasize the situated and embodied nature of intelligence. An alternative to internal world models is the 'simulation hypothesis' by Hesslow [2002] which accounts for the 'inner world' in terms of internal simulation of perception and behavior. Inspired by these observations, we are developing the Internal Rehearsal System (IRS) which utilizes one type of internal simulation of perception and behavior.

5.1 Design of The Internal Rehearsal System (IRS)

Internal simulation research has now moved into the robotics field [Shanahan, 2005]. Shanahan's architecture, shown in Figure 14, involves two separate loops, the reactive or first-order loop and the cognitive or higher-order loop. The first-order loop involves the sensory cortex (SC), motor cortex (MC), and basal ganglia (BG). This loop directly maps sensory input to motor actuation. The higher-order loop internally rehearses the decision from the first-order loop and changes the output of the system based on the observation of this rehearsal through the Amygdala (Am) or emotion system.

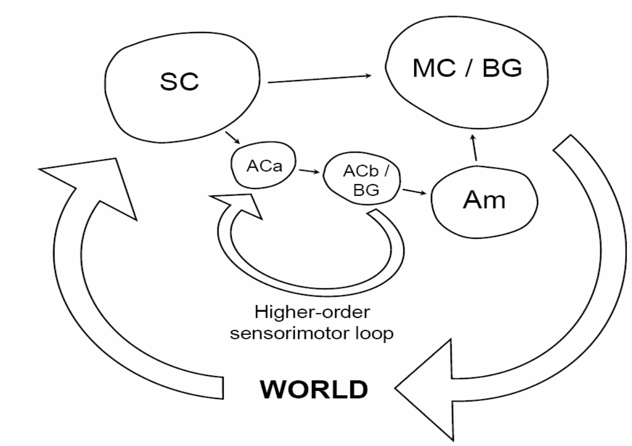


Figure 14. Shanahan's Cognitive Architecture with Internal Simulation [Shanahan, 2005]

In our architecture, the Self Agent handles dual sensory-motor loops in Shanahan's model for task execution as shown in Figure 15. The First-Order Response Agent (FRA) is responsible for the reactive and routine responses of the system while the Central Executive Agent (CEA) is responsible for the cognitive response. The Internal Rehearsal System (IRS) takes the working memory chunks as the motor commands, the current situation as the external state and sends a rehearsed result to CEA. If IRS produces a poor prediction, CEA will suppress the Activator Agent, replace the working memory chunks, and tell the Activator Agent to switch action.

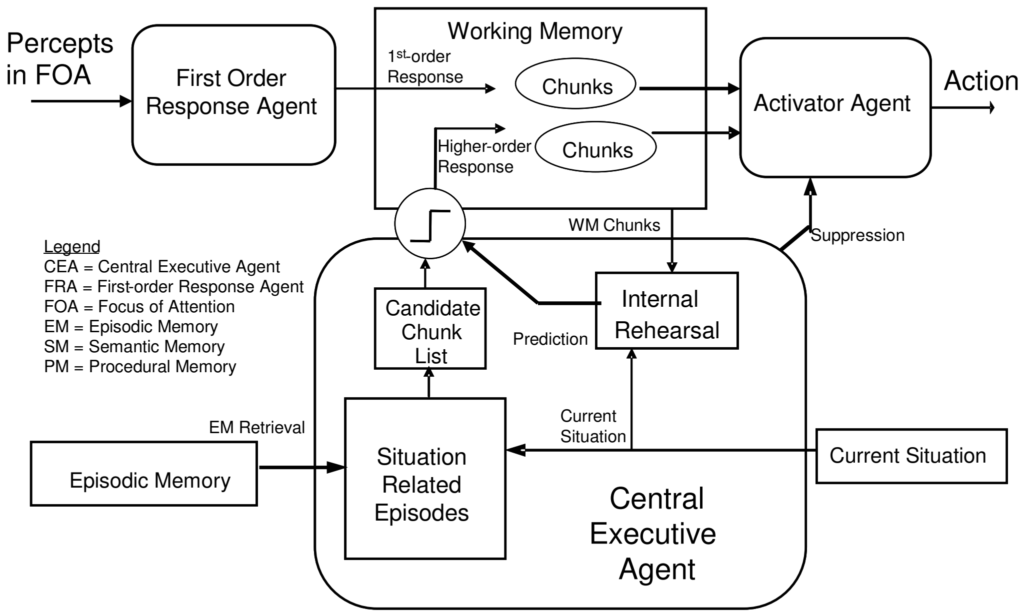


Figure 15. ISAC Self Agent Cognitive Cycle

5.2 Collision Detection

IRS uses collision detection techniques described in [Charoenseang, 1999]. This work involved the use of virtual spheres around each joint of the robot which provided virtual force feedback on contact as shown in Appendix 6. These spheres were designed to prevent both arms from colliding with an object or each other. The key to the spherical collision detection method is the distance from the center of one sphere to another. A collision occurs when two spheres touch, and if a collision occurs with the right or left arm, the arm physically moves so that the colliding spheres no longer touch.

IRS uses this collision sphere concept in a different manner. Instead of using spheres on the elbow, wrist and end effecter of the robot, the percept on the Sensory Egosphere is seen as a sphere. Figure 16 shows the ISAC Simulator where one object, Barney doll, is shown on the SES as a sphere in (x, y, z) coordinates. If either arm in the simulator collides with this sphere, a collision has occurred. The simulated arm has 6 collision detection points located in the shoulder, bicep, elbow,

forearm, wrist, and end effector of each .

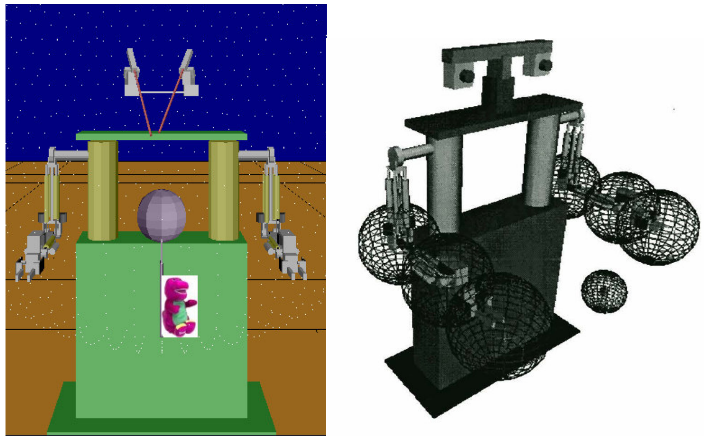


Figure 16. ISAC Simulator Displaying a Barney Doll as a Sphere

When IRS is invoked by CEA, it takes the current behavior chunk as the motor command and the current environment ISAC is in as the current state. After CEA selects a behavior to perform the skill described by the task, IRS internally rehearses the behavior with the percept corresponding to the current state using an accelerated Verb/Adverb Interpolation technique. This acceleration is performed by using only fourth of the interpolated points with Verbs and Adverbs to speed up rehearsal. If a collision occurs with the percept during the rehearsal, IRS returns the percept, the step in the Verb/Adverb interpolation where the collision occurred, and the total number of joint steps in the interpolated motion to CEA.

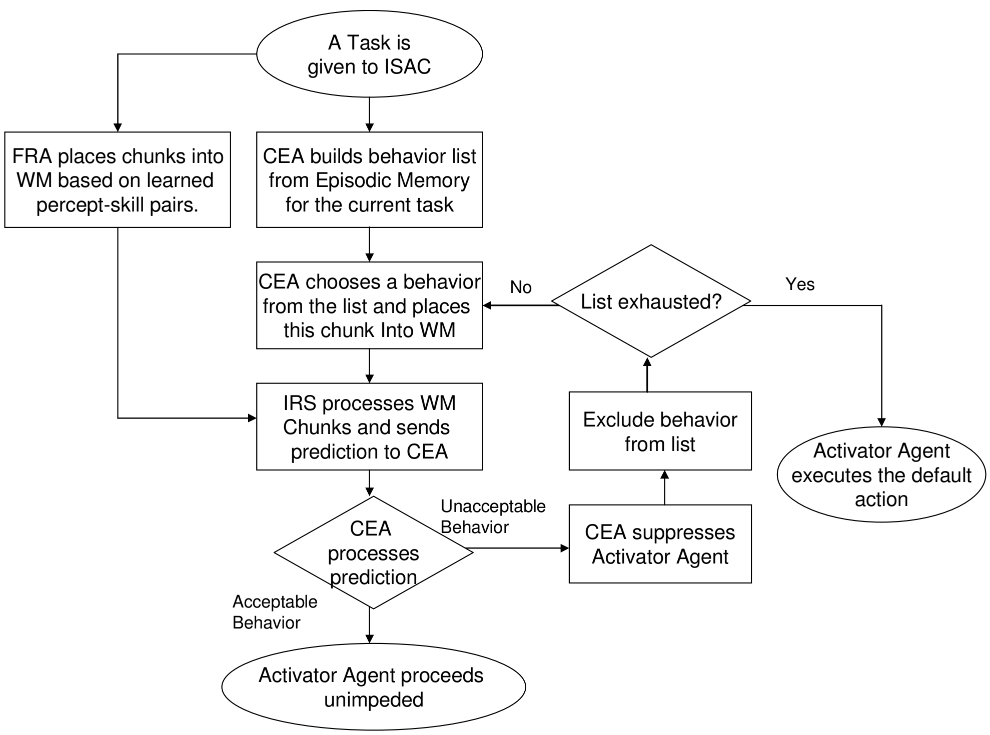


Figure 17. Flow Chart of CEA Execution involving Internal Rehearsal

Figure 17 illustrates the internal rehearsal process. First, CEA will a behavior list based on past episodes involving the given task. At the same time, FRA will determine if the percept has a corresponding routine learned object-skill pair from LTM; if it does, FRA will immediately place the corresponding percept as chunks into the working memory, runs internal rehearsal, and send the result to CEA. CEA will use the prediction from IRS to determine if t he Activator Agent should be suppressed or not. If CEA determines an acceptable, it will remove this behavior from the list and select new chunks. This process continues until a suitable skill is found to accomplish the task or the list is exhausted.

5.3 Internal Rehearsal Experiment

The following experiment is designed to evaluate how FRA, CEA, and IRS work together. The experiment involves two percepts: Barney (goal percept) and the Lego toy (obstacle percept). The experiment will proceed as follows:

1. A task to *reach to Barney* is given to ISAC. FRA immediately places ReachRight and Barney into the working memory (WM) as chunks.
2. Using the chunks, IRS will try to reach to the Barney with the right arm using an accelerated Verb/Adverb interpolation, but predicts a collision with the Lego toy.
3. CEA will suppress the Activator Agent based on this prediction from IRS.
4. CEA will use the episodic retrieval technique as shown in Appendix 5. CEA will replace the chunk ReachRight to ReachLeft.
5. IRS will reach to the Barney with the virtual left arm. This reach will be successful and reach the Barney percept.
6. CEA will let the Activator Agent proceed to reach the Barney using the left arm.

5.4 System Performance

During this experiment, ISAC perceives a Lego toy and the Barney doll, and these two percepts are posted on the SES. The ISAC Simulator displays both the position of the object on the SES and its (x,y,z) coordinates as two spheres. This can be seen in Figure 18.

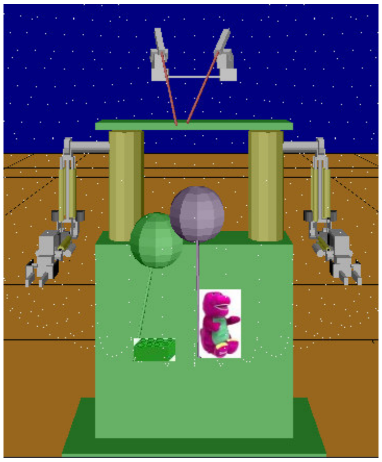


Figure 18. ISAC Simulator with Two Objects during the Experiment

ISAC was given a command to reach to the Barney. At this time, FRA placed two chunks “ReachRight” and “Barney” into working memory. (ISAC is right handed.) Both the Activator Agent and IRS began to process these chunks. IRS completed the computation within 3.202 seconds and sent its results to CEA. At the same time, the Activator Agent sent a motion command to the Right Arm Agent to perform the reaching motion. The Right Arm Agent would take 10.985 seconds to perform this type of reach if no obstacle exists.

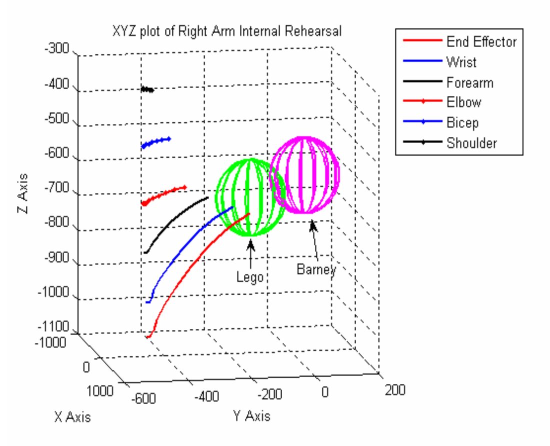


Figure 19. Right Arm Internal Rehearsal

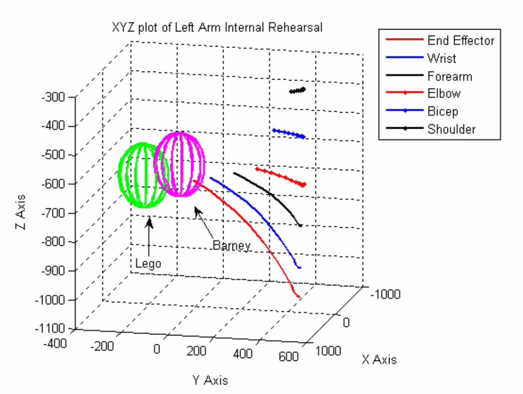


Figure 20. Left Arm Internal Rehearsal

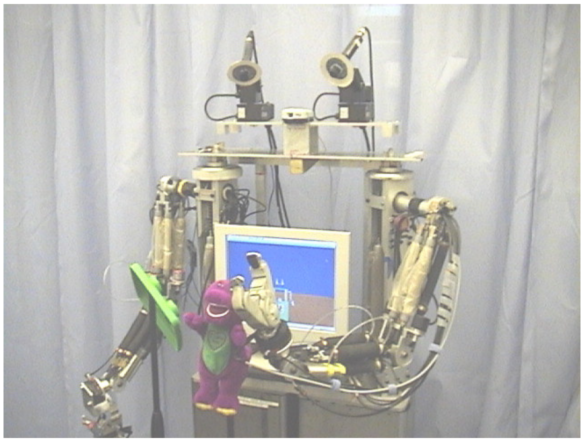
When IRS finished, the following output was sent to CEA: [15 68 lego_toy]. This means that during the simulation, IRS determined a collision with the Lego toy in the fifteenth step of the verb/adverb interpolated reach behavior out of the total of 68 interpolated steps. Figure 18 shows the trajectories of the right arm collision points during the rehearsal. CEA took this result and determined that it did not reach to Barney. CEA then suppressed the Activator Agent and prevented the right arm from further action.

CEA then decided to use an episode involving the left arm and replaces the working memory chunks with “ReachLeft” and “Barney”. IRS and Activator Agent were once again initiated, and IRS internally rehearsed the reach skill and determined no collision with the Lego toy. Indead, IRS found a collision with the Barney, a success, as shown in Figure 20. Both the wrist and end effector points entered the Barney percept sphere on the sixteenth step of the Verb/Adverb interpolation. The output of IRS wass [16 69 barney_toy] after 2.983 seconds. CEA determines this as a success and did not impede the Activator Agent thus allowing ISAC to reach to Barney using his left arm. The result of the experiment is shown in Table 2 and Figure 21.

Performance Time (Seconds)		
	Right Arm	**Left Arm**
Internal Rehearsal	3.202	2.983
Arm Agent	10.985	7.110

Table 2. Performance time

The Internal Rehearsal System currently determines a collision by using collision spheres. Collision spheres are used since most of the objects ISAC encounters such as dolls and bean bags are small in size and can be safely represented by spheres. In the future, IRS need to include larger objects such as tables or chairs as shown in Figure 20.



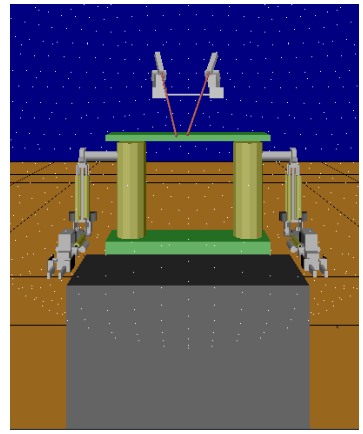


Figure 19. ISAC Reaching to Barney with Left Arm Figure 20. Collision with a table

6. A SELF-MOTIVATED, INTERNAL STATE-BASED ACTION SELECTION MECHANISM

In his book, Damasio [1999] associates higher-order reasoning with planning and conscious behavior execution. Below this level of reasoning is unconscious feelings, emotions, and basic life regulations (homeostasis). It is the interaction of these various levels that enable cognitive systems (such as humans) to handle many complex situations. Embedded cognitive systems such as cognitive robots may also face situations where they cannot rely on conscious state alone to make appropriate decisions. Therefore, in our architecture we are developing a subsystem capable of lower-level unconscious, internal state-based responses exhibited by biological cognitive systems. In this section we discuss the use of one such internal state *affect* in our work. In our architecture, affect is maintained by the Affect Agent, which keeps track of the current affective level of ISAC. Similar to the work of Shanahan [Shanahan, 2006], affect interacts with CEA by running in parallel, influencing focus of attention, and mediating task execution by influencing the probabilistic decision making model within CEA [Ratanaswasd, et al, 2006]. It is important to note that while the affect system we are developing contains similarities to human affect, it is not grounded and thus it should more appropriately be considered as an "affect-like" system. For brevity, we refer to it as an affective system.

As seen in earlier sections, the Attention Network functions as an important part of cognitive control loops within the ISAC cognitive system. In the following, we will explore the manner in which internal states such as emotions interact with attention and the rest of the cognitive system. This section involves the Affect Agent, the Intention Agent, and CEA.

6.1 Role of Internal State

The role of internal states in goal and action selection has been explored by a number of researchers. For example, Cos-Aguilera, et al., and Canamero focused on the role of internal homeostatic variables within the robotic system [Cos-Aguilera, et al., 2005; Canamero, 1997]. Such variables are used to indicate a preferred normalcy (i.e. homeostasis) within the robot. For example, a mobile robot may possess the homeostatic variable, *hunger*, which keeps track of the remaining battery life of the robot. As the value of this internal variable lessens, an urge (or drive) motivates the robot to seek out some form of refueling station. Using a set of homeostatic variables, coupled with appropriate drives, robots can attempt to maintain an internal level of homeostasis during task execution. Cos-Aguilera and Canamero both used this type of system to learn behavior selections.

Internal states have also been used to create robotic emotional systems [Gockley, et al, 2006] [Breazeal, 2004]. One of the primary goals of robotic emotional systems is to enable a more natural interaction between human and robot. We hypothesize that in order for such robotic systems to be truly effective and useful, the robot must use internal states in its decision making and action selection in a manner that is more readily apparent to a human counterpart.

Our main goal in using internal state-based action selection and decision making is to develop a dynamic system capable of acting based on internal preferences that can be modified through experience. In this section, the use of internal variables that represent the system's deviation from a normative emotional level is proposed. While it is not sufficient for a cognitive system to use only emotional preferences in the decision-making process, we believe that it is nonetheless important for any cognitive system to have emotions.

The total state of the ISAC cognitive system is represented by two sets of state variables, external and internal. External state variables, S_{ext} , include percepts such as detected objects, faces, keywords, sounds, etc. Internal state variables, S_{int} , include hardware parameters such as joint angles, positions of right and left hands as well as qualitative variables such as intention and affect (Figure 2). These variables combine to form the overall situation:

$$Situation\ (S_{total}) = S_{ext} \times S_{int} \qquad (2)$$

The work discussed in this section requires S_{ext} and a portion of S_{int}. The portion of S_{int} used, denoted as S_{affect} , represents the internal variables used by the Affect Agent. S_{affect} is not designed to be a constant. Rather, past experience (e.g., previous reward) should be used to update the preferences to a novel situation over time. In this manner, S_{affect} is used to reflect the both the system's innate preferences for a situation as well as reflecting the system's dynamic state change. The past experience comes in the form of Episodic Memory (see Appendix 7), a form of memory currently being developed for ISAC.

6.2 Affect Agent and Excitement Meter

Robots operating in the real world often encounter situations where more than one choice of action could be considered acceptable. In any cognitive robot, this choice should be mediated by past experience and internal states. This mediation enables a cognitive robot to make its own choices to deal with competing demands. Towards the goal of developing a self-motivated action selection mechanism, we have begun to develop a means of allowing ISAC to make decisions based on its own preference. In our case this preference is represented through affect variables associated with particular tasks. Though much prior work involving affective states in the decision-making process involving artificial systems has been conducted [Gockley, et al, 2006] [Picard, et al, 2004], there does not seem to exist a formal, universally accepted definition of what *affect* is. In our work, we adopted the working definition that "affect is an internal unconscious level of state, including feelings and emotions, in response to a particular situation or event". This is similar to the use of the term in [Gockley, et al, 2006]. When cognitive content is attached to emotions, (e.g., *playing with toys* is exciting) then this definition becomes similar to that used in [Picard, 1997] and [Franklin, et al, 2006].

6.2.1 Calculation of Excitement

In the ISAC architecture the Affect Agent determines affective responses to the current situation. These responses implicate which choices lead to higher or lower affective levels. This is similar to the Winner-Take-All (WTA) behavior selection technique discussed in [Ho, et al., 2005]. In their approach, an agent has a set of homeostatic variables that need to be maintained. The motivation to compensate each of these variables is determined at each time step, and the behavior corresponding to the homeostatic variable with the most motivation is chosen. Their study shows that the WTA strategy resulted in longer agent life spans (over all settings) when compared with Static-Thresholding and Voting-Based selection techniques. There is an important difference however between the WTA behavior selection technique and the work discussed here. Their work revolves around internal variables that are homeostatic or "life-sustaining" (i.e. *hunger*, *thirst*, *fatigue*, etc.). The work discussed here involves "preferential" internal variables. This difference is analogous to an agent needing to eat (i.e. life-sustaining drive) versus preferring to eat pizza rather than asparagus (i.e. preferential drive). Choosing the least preferred option (asparagus) does not kill the agent.

Because of this difference, the behavior selection mechanism we propose is a modification of the WTA technique where the winning motivation has the highest probability of taking all. Based on the affective response to the situation, the Affect Agent modifies the probabilities used by the CEA for action selection (Appendix 5). This happens during the following process: CEA determines initial probabilities for choosing each task; these values are passed to the Affect Agent where they are modified based on the current affective responses; the new probabilities are returned to CEA along with the strength of the individual affective responses. Based on the overall strength of the affective responses, CEA may choose to ignore or accept the updates in probabilities determined by the Affect Agent. For example, the affective response to the event "person leaving the room" may be very low and CEA may choose to disregard any change in the probabilistic model based on this event. However, the event "person entering the room with toys" may have a very high affective response and the system should more likely switch tasks.

Appendix 5 discusses the decision-making model used by CEA. With affect variables added to the system, the action associated with the higher affective state has a higher priority, and subsequently priorities of all other actions are decreased. This affect-based dynamic prioritization of an action is used to determine the probability of the chosen action.

Figure 22 illustrates this process. The Affect Agent monitors the current situation, i.e. the external states S_{ext}. The Affect Agent then determines new values for affective variables to be used on the next time interval. CEA is simultaneously monitoring the situation and creates an action list from similar episodes in the Episodic Memory, and assign probabilities to the actions.

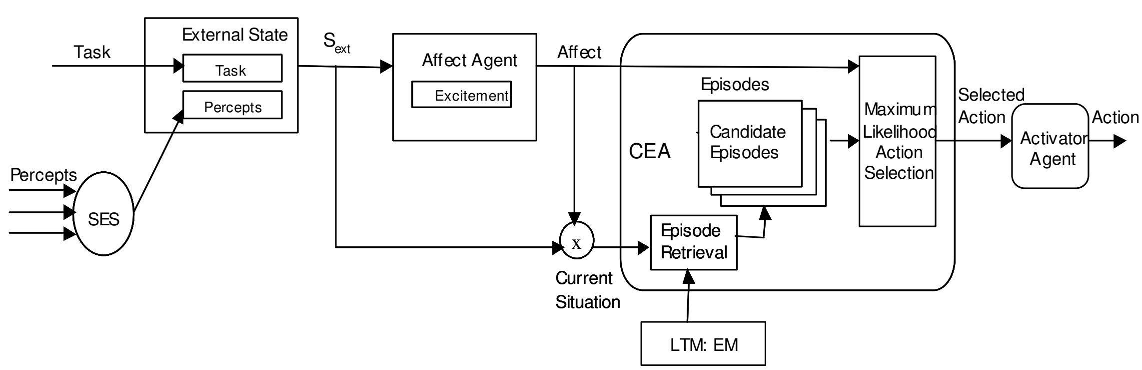


Figure 22. Affect-based Dynamic Prioritization of Action Selection

Prior work involving the Affect Agent used fixed values [Ratanaswasd, et al., 2006]. The current model extends to a dynamic affective variable called *excitement*. Our model of excitement uses the following function as suggested by Picard [Picard, 1997] for determining the affective level:

$$\text{Excitement level} = Ae^{-Bt} \quad (3)$$

where A and B are calculated as follows: For new situations, the initial values of A and B (A_{init}, B_{init}) are set to the system's innate preference values. Once a situation has been encountered and the experience stored within the Episodic Memory, the values (A, B) can be recalled from that episode. As ISAC gains more experience with a particular situation, A and B must be updated. Following summarizes our tentative thinking of how to update the values.

It is important that the excitement increase due to the successful past experience. However, encountering the same situation frequently will have an adverse effect on the excitement. Thus we are thinking of using the following update equations for the affect parameters:

$$A_c = f(\alpha, \beta, A_{ret}) \quad (4)$$

where A_{ret} is the retrieved value of A. The parameter α is a bounded function of the previous success of the situation and β is function of the number of occurrence within the Episodic Memory of the retrieved situation. For this system to function properly, it is important that α be bounded and that the bound of α not exceed the limit of β. This ensures that A will not grow without bound. The sigmoid function, $1/(1+e^{-x})$, is used for α where 'x' is the success of the previous experience. The function, log(y), is used for β where 'y' is the number of occurrences of the given situation. Using this design, the excitement associated with any situation is a function of that situation's initial excitement level, A_{init}, the system's recorded success (or failure) with the situation, and the frequency of occurrence of the situation. Using a functional design with the bounded parameter, α, in the numerator and β in the denominator ensures that eventually, repetition will win out and through continued exposure to a situation the excitement ISAC associates with that situation will return to neutral. The calculation of B_c is done in a similar fashion except that the signs are reversed.

$$B_c = g(\alpha, \beta, B_{ret}) \quad (5)$$

Figures 23 and 24 show how the values (A_c, B_c) change as a function of the number of trials performed where the success of the past experience is assumed to be maximum over all trials. These figures represent the basic equations without the use of any adjustment constants. Because no adjustments are made the scale shown is an arbitrary one.

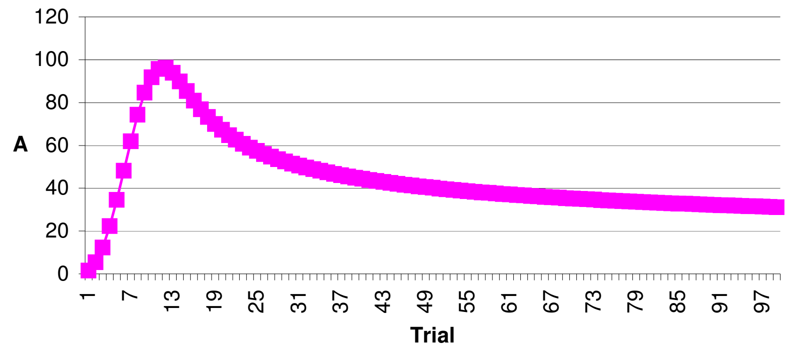


Figure 23 Evolution of Parameter A over time with a Constant Positive Reward

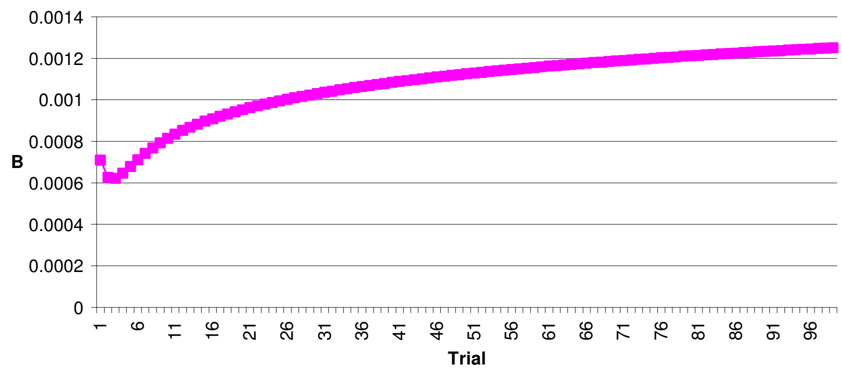
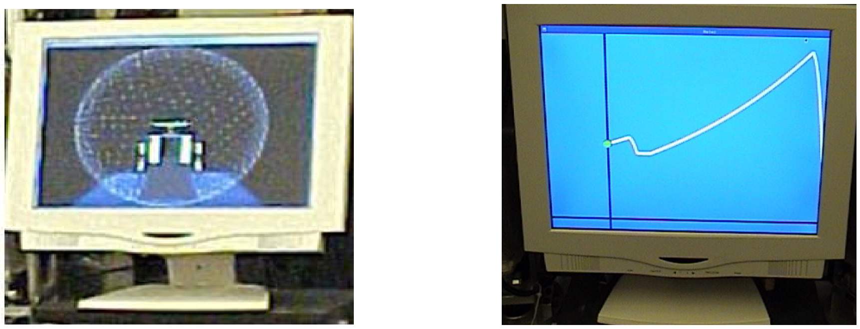


Figure 24 Evolution of Parameter B over time with a Constant Positive Reward

6.2.2 Excitement Meter

In parallel to the theoretical development of the affective variables, we are working on a graphical display method of excitement called the *Excitement Meter*. The main reason for the development of the Excitement Meter is to aid human-robot interaction. As the internal states of ISAC grow, it is important to have a means of relating that information to human counterparts. This meter calculates the excitement response level to a set of stimuli and displays the history of excitement value on the "chest" monitor mounted on ISAC (Figure 25). The current value of excitement is shown on the black axis. The previous values trail off to the right. Figure 25(a) shows the chest monitor as it normally is used showing the Sensory EgoSphere. Figure 25(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.

The current implementation of the Excitement Meter is still too simple. However, a pilot study (performed by the Vanderbilt Psychology Dept.) indicates that the presence of the meter aids humans to understand why ISAC took particular actions.



(a) (b)

Figure 25. ISAC Chest Monitor

7. CONCLUSIONS

This paper outlined our effort to develop an integrated robotic cognitive architecture in which a number of cognitive loops are running in parallel. Collectively, these cognitive loops are expected to realize a certain level of cognitive control functionalities for our ISAC humanoid robot. Results of experiments conducted so far are encouraging. However, many challenges still remain in order to truly realize integrated cognitive robotic systems.

ACKNOWLEDGEMENTS

Special thanks to Flo Wahidi, and the graduate students at the Center for Intelligent Systems. This work is supported in part under the NSF grant EIA0325641, “ITR: A Biologically Inspired Adaptive Working Memory System for Efficient Robot Control and Learning” and the NSF grantIIS-0433653, "Intentional Vision in Humans and Robots”.

REFERENCES

[Anderson, et al., 2004] 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.

[Arbib, 1981] 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.

[Baddeley, 1986] A.D. Baddeley, Working Memory, Oxford: Clarendon Press, 1986.

[Botvinick, et al., 2001] 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.

[Branicky, et al., 1994] M. Branicky, “Analyzing continuous switching systems: Theory and examples”, *Proc. of American Control Conference*, Baltimore, MD, June 1994

[Breazeal, 2004] C. Breazeal, “Function Meets Style: Insights From Emotion Theory Applied to HRI,” III Transctions on SMC-Part C. Applications and Reviews, vol. 34, no. 2, pp. 187-194, May 2004.

[Brooks, 1986] 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.

[Canamero, 1997] D. Canamero, “Modeling Motivations and Emotions as a Basis for Intelligent Behavior”, *Proc. of the First International Symposium on Autonomous Agents*, 1997.

[Charoenseang, 1999] S. Charoenseang, “A PC-Based Virtual Reality System for Dual-Arm Humanoid Robot Control,” Ph.D. Dissertation, Vanderbilt University, 1999.

[Clancey, 1997] W.J. Clancey, Situated Cognition, Cambridge University Press, 1997

[Clark, 1997] Clark, A., Being There: Putting Brains, Body, and World Together Again. MIT Press, 1997.

[Pfeifer and Scheier, 1999] R. Pfeifer and C. Scheier, Understanding Intelligence. MIT Press, 1999.

[Cohen and Shoup, 1997] A. Cohen and R. Shoup., “Perceptual dimensional constraints on response selection processes,” *Cognitive Psychology, 32,* 128-181., 1997.

[Cohen and Shoup, 2002] A. Cohen and R. Shoup, “Response selection processes for conjunctive targets”, *Jo. of Experimental Psychology: Human Perception and Performance,* vol. 26*,* pp. 391-411, 2002.

[Cos-Aguilera, et al 2005] I. Cos-Aguilera, L. Cañamero, G.M. Hayes, and A. Gillies, “Ecological Integration of Affordances and Drives for Behaviour Selection”, *Proc. of MNAS2005*, Edinburgh, 2005.

[Damasio, 1999] A. Damasio, The Feeling of What Happens: Body and Emotion in the Making of Consciousness. Heinemann, London, 1999.

[Dearden and Demiris, 2005] A. Dearden, Y. Demiris, “Learning Forward Models for Robots”, in: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), 2005, pp. 1440-1445

[Franklin and Ramamurthy, 2006] S. Franklin & U. Ramamurthy, “Motivations, Values and Emotions: 3 sides of the same coin”, *Proc. of the 6th International Workshop on Epigenetic Robotics*, Paris, France, Lund University Cognitive Studies, Vol. 128, pp. 41-48, September, 2006

[Gazzaniga, 2002] M.S. Gazzaniga, Cognitive Neuroscience. The Biology of the Mind, 2nd ed., Norton & Co., New York, 2002, p. 615.

[Gockley, et al 2006] R. Gockley, R. Simmons, & J. Forlizzi, “Modeling affect in socially interactive robots”, *Proc. of 15th Annual IEEE International Workshop on Robot and Human Interaction*, London, UK. 2006.

[Goldman-Rakic, 1987] P.S. Goldman-Rakic, "Circuitry of the Pre-frontal Cortex and Regulation of Behavior by Representational Knowledge", In F. Plum and V. Mountcastle, eds., *Handbook of Physiology* pp 373-417, American Physiological Society, Bethesda MD.

[Gordon and Hall, 2006] S.M. Gordon, J. Hall, " System integration with working memory management for robotic behavior learning", *Proc .of the 5th International Conference on Development and Learning*, Bloomington, IN, June 2006.

[Hambuchen, 2004] K.A. Hambuchen, "Multi-modal attention and binding using a sensory egosphere", Ph.D. Dissertation, Vanderbilt, Nashville, TN, May 2004.

[Harnad, 1990] S. Harnad, "The Symbol Grounding Problem", Physica D 42: 335-346, 1990.

[Hesslow, 2002] G. Hesslow, Conscious thought as simulation of behavour and perception. *Trends in Cognitive Science*, 6(6). 242-247, June 2002.

[Ho, et al., 2005] W.C. Ho, O. Avila-Garcia, C.L. Nehaniv, "Assessing the performance of different behavior selection architectures in a large and complex virtual environment", *IEEE Congress on Evolutionary Computation*, 2005, Vol. 1, pp 359-365.

[IEEE Control Society] Technical Committee on Intelligent Control, IEEE Control Systems Society, http://robotics.ee.nus.edu.sg/tcic/

[Jenkins and Mataric, 2003] 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, Austrailia, July 14-18, 2003.

[Kawamura, et al., 2004] 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.

[Lee and Thompson, 1982] D.N. Lee and J.A.I Thompson, "Vision in action: The control of locomotion," in D. Ingle, M.A. Gooddale & R.J.W. Mansfield (eds), *Analysis of Visual Behavior*, MIT Press, 1982.

[Miller, et al., 1996] 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.

[Miller, 1956] Miller, G.A., The Magical Number Seven, Plus or Minus Two: Some Limits in Our Capacity for Processing Information. *Psychological Review*. vol. 65. pp. 81-97.

[Miller, 2003] Miller, E.K., Cognitive Control: Understanding the brain's executive, in Fundamentals of the Brain and Mind, Lecture 8, June 11-13, MIT Press, Cambridge, MA, 2003.

[Pack, et al., 1997] 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.

[Peters, et al., 2001] 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.

[Phillips and Noelle, 2005] J.L. Phillips, D. Noelle, "A biologically inspired working memory framework for robots", *Proc. of the 27th Annual Conf. of the Cognitive Science Society*, 2005, pp. 1750-1755.

[Platt, et al., 2006] R. Platt, R. Grupen, A. Fagg, "Improving Grasp Skills Using Schema Structured Learning", *International Conference on Development and Learning*. Bloomington, Indiana. May, 2006.

[Picard, 1997] R.W. Picard,. Affective Computing, MIT Press, Cambridge, MA. 1997

[Picard, et al 2004] R.W. Picard, S. Papert, W. Bender, B. Blumberg, C. Brazeal, D. Cavallo, T. Machover, M. Resnick, D. Roy, and C. Strohecker, "Affective Learning – A Manifesto", *BT Technology Journal*, Vol. 22, No. 4, pp. 253-269, October 2004.

[Ratanaswasd, et al., 2006] 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.

[Rose, et al., 1998] 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.

[Shapiro, 1971] S. C. Shapiro "A net structure for semantic information storage, deduction and retrieval," *Proc. IJCAI-71*, 1971, pp. 512-523.

[Shanahan, 2006] M.P. Shanahan, "A cognitive architecture that combines internal simulation with a global workspace", *Consciousness and Cognition*, vol. 15, 2006, pp. 433-449.

[Spratley, 2006] 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.

[Sun and Scassellati, 2004] G. Sun and B. Scassellati, "Reaching through learned forward model" in *IEEE-RAS/RSJ International Conference on Humanoid Robots*. Santa Monica, CA: IEEE, 2004.

[Sutton, 1988] R.S. Sutton, "Learning to Predict by the Methods of Temporal Differences", *Machine Learning*, 3:9-44, 1988

[Thrun and Pratt, 1998] S. Thrun and L. Pratt, eds., Learning to Learn, Kluwer Academic Publishers, Boston, MA, 1998

[Waugh and Norman, 1965] N.C. Waugh and D.A. Norman, "Primary Memory", *Psychological Review*, 72:89, 104, 1965

[Wolpert and Kawato, 1998] D.M. Wolpert and M. Kawato. "Multiple Paired Forward and Inverse Models for Motor Control", *Neural Networks,* 11, 1317–1329, 1998.

[Ziemke, et al., 2002] T. Ziemke, *et al.,* Blind Adaptive Behavior Based on Internal Simulation of Perception. Technical Report HS-IDA-TR-02-001, Computer Science Dept., University of Skövde, 2002.

APPENDICES

Appendix 1. Working Memory System (WMS)

1.1 Working Memory Toolkit - WMtk

The WMtk is an ANSI C++ library of classes and methods that can be used to instantiate and train WMS. The WMtk loads "chunks" of information in the form of simple data structure from a supplied list of candidate chunks. These chunks are void data structures that represent stored information within the memory in the system. In our system, the candidate chunk list is derived from filtering the contents of STM (SES), LTM, and the Episodic Memory. The WMtk encodes the current state of the system within a state feature vector in the toolkit, compares this with a chunk feature vector and then chooses chunks for which it expects the most reward. To make these decisions, the state and chunk feature vectors are passed through a neural network that learns using the temporal-difference (TD) learning algorithm [Sutton, 1988]. In Figure 3, the *Learned Network Weights* represent the particular instance of the WMtk used for the current situation.

Because the WMtk uses feature vectors and neural nets, some of the same computational constraints associated with machine learning are present. For instance, if new states are added to the system after training, WMS must be re-trained. In order to for WMS to be scalable, the system requires the ability to dynamically create new instances of the WMtk for novel situations. As discussed in Section 3.2.1 separate instances of the WMtk are used for different category tasks.

1.2 Memory Contents during WM Training

Table A1 shows the contents of short-term (SES) and long-term memory (LTM) during the experiment discussed in Section 3. In the experiment, two bean bags were present in front of ISAC. Additionally, five behaviors had been trained and placed in LTM. This information was encoded into the working memory as "chunks," i.e. void data structures. The WMtk then chose two from these chunks and the contents of working memory were used for task execution. In other words, if the chunks *Behavior5* and *red_bag* were present, ISAC performed the fifth behavior in LTM on the red bean bag percept, i.e. the two-dimensional reach motion towards the red bean bag. Table A2 shows sample contents of working memory during four of the training trials. When a percept and/or a behavior chunk were not present, the missing chunk(s) were filled in at random.

SES	*LTM*
1. Percept-1, Semantic label= "blue_bag", Location = *node*1518	1. Behavior-1
2. Percept-2, Semantic label = "red_bag", Location = *node*1212	2. Behavior-2
	3. Behavior-3
	4. Behavior-4

	5. Behavior-5

Table A1. Memory Contents During Simulation Training

	Working Memory Contents			
Sample Trial #:	1	2	3	4
Chunk 1	blue_bag	red_bag	*Behavior-3*	Behavior-5
Chunk 2	Behavior-2	Blank	blue_bag	red_bag
Random:	NA	Behavior-1	NA	NA
Reward:	50.7	0	0	22.4

Table A2. Working Memory Contents During WMS Training

APPENDIX 4. Verbs and Adverbs Algorithm for Behavior Execution

Verbs and Adverbs is a motion interpolation technique originally developed for computer graphics [Rose, et al., 1998]. We use this technique because it is scalable to work on an arbitrarily sized dimension spaces. 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* [Rose, et al, 1998] [Spratley, 2006]. 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. These keytimes relate to the structure of the motion and are conceptually similar to breakpoints of the motion. Figure A1 shows the structure for three example motions. The example motions shown are recordings 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 [Jenkins and Mataric, 2003]. 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.

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 body that the particular motion is to extend. To extend this method to include other features, such as speed, example motions representing the variability of these features need to be added to the verb. In addition, new adverbs also need to be added.

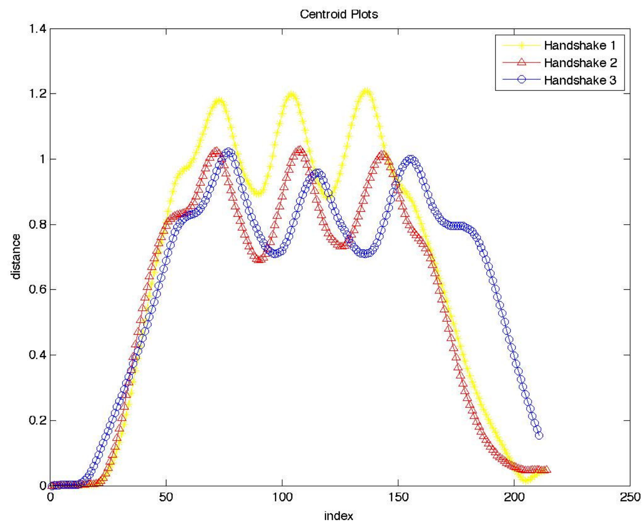


Figure A1. Example Motion Structures Using Kinematic Centroid [Spratley, 2006]

For our experiments, motions were provided to the system through teleoperation. Groups of exemplar motions were used to create verbs. Each verb directly corresponds to a skill, as discussed in Section 3, and each verb is given a label (such as Behavior1, Behavior2, etc.). LTM is used to store the verb exemplars and the adverb parameters for each verb. New motions for the behaviors that represent *reaching* or *handshaking* are interpolated at run time using the new (desired) adverb values. This approach is analogous to a memory-based learning approach mentioned in [Thrun & Pratt, 1998]. One important limitation in the Verbs and Adverbs technique is that new motions are never extrapolated. This is due to the fact that extrapolated motions can potentially lead to undesirable or unachievable arm configurations. When extrapolation is required, the necessary behavior is approximated through interpolation at the adverb limit.

Appendix 3. Focus of Attention and Action Selection

Humans pay attention by emphasizing the locations of percepts with high saliency. This process is known as *spatial attention* [Cohen and Shoup, 1997]. In our architecture, spatial attention is realized through assigned *Focus of Attention* (FOA) by the Attention Network [Hambuchen, 2004]. The Attention Network puts flags on percepts with high saliency as input from the Affect Agent. The flagged percepts become candidates which to be posted to the working memory for further processing.

During an event, such as when ISAC is given a task, a set of past episodes are recalled from the Episodic Memory using cues such as percepts in FOA and task information. If no relevant percepts exist or no relevant episodes can be recalled, ISAC cannot perform this task, in which case the default action will be executed. Currently ISAC will say “I cannot do it” when it cannot perform a task. In case that multiple tasks are presented, relevant past episodes for all tasks are recalled and ISAC must decide which task will be handled first. This decision is made based on the utility values of the tasks. Currently, the utility value is computed using the success rate of the task, referred to as the expected reward as follows: For a task T_k, assume N_k episodes are retrieved. Each episode receives the score S_i=1 if it was a success, otherwise -1. The expected reward R_k for the task is computed by

$$R_k = \frac{\sum_{i=1}^{N_k} S_i}{N_k} \qquad (1)$$

Once a task is selected, a behavior must be selected. A list of behaviors is created from the recalled episodes. The list is arranged according to the priority value of the task as follows: Let B_j be a behavior within the list and E_{Bj} be the set of recalled episodes that contain B_j for the task T_k. The priority, p_j , then, is computed as

$$p_j = \frac{S_j}{M} \qquad (2)$$

where M is the number of members in E_{Bj}, and S_j is the summation of relevancy of successful episodes in E_{Bj}.

The relevancy of an episode is computed based given the current task and the percepts by

$$r = \frac{1}{(d_p + 1) \times (d_b + 1)} \qquad (3)$$

where,

d_p is the number of edges on the SES geodesic dome from the node that the current percept appears to the percept present within the episode, and

d_b is the traveling distance of the behavior B_j when it was used for the past execution.

Once the behavior list is generated and rearranged, the behavior from the top of the list is selected for the task execution. The flow chart in Figure A2 summarizes this behavior selection process.

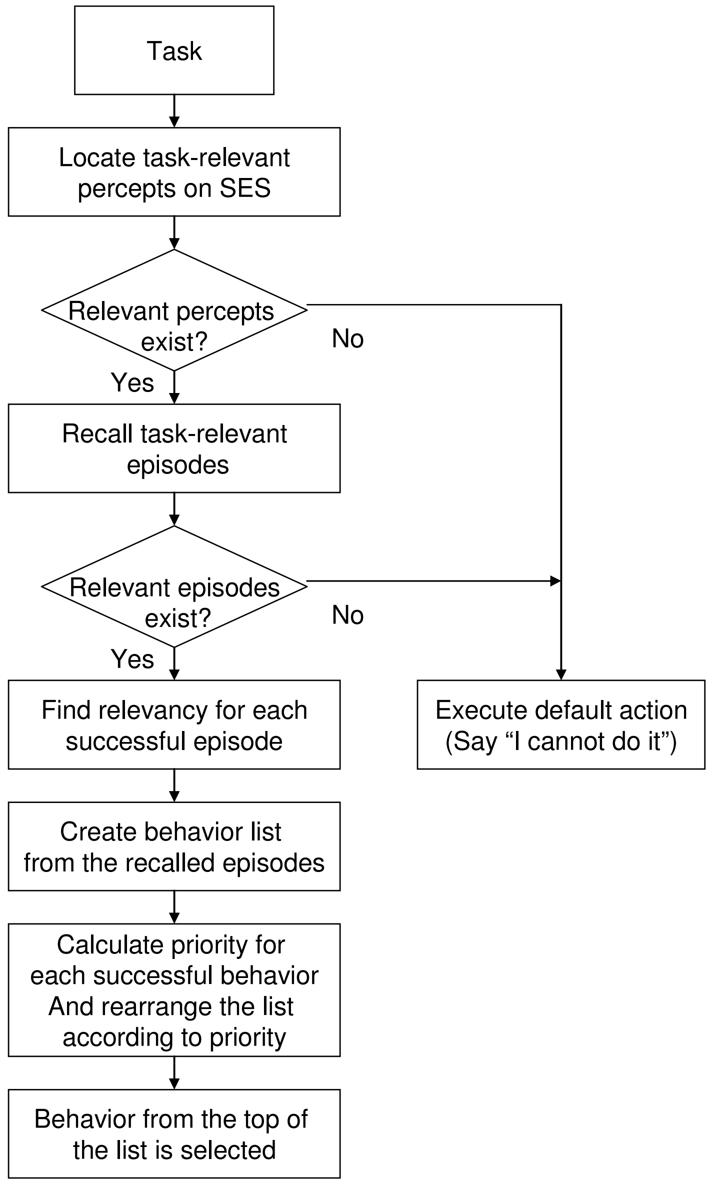


Figure A2: Summary of the behavior selection process

Appendix 4. Episodic Memory – Under development