SANDIA REPORT
SAND2003-4153
Unlimited Release
Printed December 2003

Final Report for the Endowment of Simulator Agents With Human-Like Episodic Memory LDRD

Chris Forsythe, Carl Lippitt, Patrick Xavier, David Schoenwald, Ann Speed, Mark Schaller, and Edward Thomas

Prepared by Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-4153
Unlimited Release
Printed December 2003

Final Report for the Endowment of Simulator Agents With Human-Like Episodic Memory LDRD

Chris Forsythe
Computational Initiatives Department

Ann Speed
Reliability Assessment and Human Factors Department

Carl Lippitt
Computational Initiatives Department

Mark Schaller
Computational Initiatives Department

Patrick Xavier
Intelligent Systems Principles Department

Edward Thomas
Independent Surveillance Assessment and Statistics Department

David Schoenwald
Evolutionary Computing Department

Sandia National Laboratories
PO Box 5800
Albuquerque, New Mexico 87185-1188

Abstract

This report documents work undertaken to endow the cognitive framework currently under development at Sandia National Laboratories with a human-like memory for specific life episodes. Capabilities have been demonstrated within the context of three separate problem areas. The first year of the project developed a capability whereby simulated robots were able to utilize a record of shared experience to perform surveillance of a building to detect a source of smoke. The second year focused on simulations of social interactions providing a queriable record of interactions such that a time series of events could be constructed and reconstructed. The third year addressed tools to promote desktop productivity, creating a capability to query episodic logs in real time allowing the model of a user to build on itself based on observations of the user's behavior.

THIS PAGE INTENTIONALLY LEFT BLANK.

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

1. Introduction

The Endowment of Simulator Agents With Human-Like Episodic Memory LDRD was undertaken to develop technology that would allow machines to emulate human episodic memory capabilities. Specifically, through this capability, a machine stores a record of its experiences that may later be recalled using various retrieval cues. This technology is differentiated from other similar capabilities (e.g., Shastri, 2000; Shastri, 2001) due to our deliberate efforts to model the functional characteristics of human episodic memory.

For many applications, there is no benefit in machine-based episodic memory emulating its human counterpart. However, our interests have focused on three significant exceptions. First, with simulation for training and analysis, it is desirable that synthetic representations of human entities behave in a realistic manner. Arguably, this entails basing the cognitive processes of synthetic entities on psychologically plausible models of human cognition, including the role of past experience in shaping ongoing cognitive processes and behavior. Through the current project, we have demonstrated a capability to generate synthetic entities that each possesses a unique set of life experiences that provide the basis for their interpretation of ongoing events.

The second application involves "Cognitive Systems" technologies. These are systems that rely on highly realistic representations of human cognition as a basis for their interactions with users. Much in the same way that cognitive processes enhance the abilities of humans to communicate and collaborate, the goal is to enable machines to similarly interact with humans on a cognitive level. For example, episodic memory provides a record of experiences that we have shared with different individuals. Consequently, we may later use that shared experience to place current events within a meaningful context (e.g., the plumber explains that this is the same problem he fixed with your hot water heater three years ago). Through the current project, we have demonstrated a similar capability for a machine to store a meaningful representation of its experiences with a given user and apply this shared experience in its subsequent interactions with the user.

Finally, early results of models already built and validated indicate that this general cognitive architecture, when populated with appropriate knowledge from an individual, can come to the same conclusions that individual does around 90 percent of the time (Forsythe et al., 2002; Jordan et al., 2002). Despite early success with this human emula-tion technology, there exists a major obstacle to its wider application to real-world problems. That obstacle is the fact that, once a cognitive model is created of an individual user or expert, that model is static. In its current form, the model is analogous to a photograph—it remains forever the same while the subject of the photograph continues to change and evolve. Therefore, in order for the emulation technology to continue to be useful to the emulated individual for a substantial period of time, the emulation technology must be able to change and evolve in parallel with the individual. That is, it must be able to learn. Therefore, the goal for the third year of this LDRD was to add to the basic cognitive framework a rudimentary learning mechanism based on episodic memory. To our knowledge, there is no other cognitive model that has an ability to learn new abstract versions of situations from multiple specific examples (cf. Holyoak and Hummel, 2001). Therefore, this capability will be a significant addition to the general capabilities of cognitive architectures in general.

The project described in this report has successfully met its objectives. When the project began in October of 2000, we had developed and demonstrated an initial capability to computationally model the cognitive processes underlying human situation recognition (Forsythe, 2001). In a rudimentary manner, this model depicted the processes whereby events are interpreted through the recognition of familiar patterns of cues within the environment. Over the past three years, this model has undergone three substantial upgrades to provide greater depth and breadth of function, including the capabilities for episodic memory described in the current report. Presently, due to this project, we now have a capability to develop psychologically plausible cognitive models tailored to specific applications, whether intelligent machine or synthetic humans, equipped with capabilities to meaningfully store a representation of their experiences and retrieve and apply knowledge of past experiences to interpret ongoing events.

As an overview, the conceptual design of the cognitive model and the episodic memory will be presented. Then, the work performed for each fiscal year of the project will be presented. Finally, a comparison of the capability resulting from these three years of work will be made with several similar technologies, including one developed by Creo Associates, called Six Degrees, along with additional ideas for future functionality of the Episodic Memory.

2. Conceptual Design

A key achievement of this LDRD has been the advancement of our capabilities for computationally modeling human cognition. This includes substantial refinement of the initial baseline model and the incorporation of additional features essential to a plausible representation of human cognition.

At this point, it is important to note that while the human being is the "gold standard" by which we measure the effectiveness of our modeling efforts, we are not attempting to model human cognition for research purposes. The implication of this statement is this: we will not hobble the computer in the interest of most realistically modeling cognition. We will, instead, make the computer as human-like as possible without removing its data-processing and data-search abilities. For example, we know that humans do not serially search through every memory they have looking for information—reaction times do not support this theory. However, the computational representation of episodic memory described here does search every record. Granted, it retrieves information in a human-like way in terms of pattern recognition, but we are not hobbling its perfect record of past events in the interest of replicating human response times to episodic memory tasks. To state this point differently, our goal is to augment human cognition—to create a man-machine symbiosis such that each takes advantage of the other's strengths. In order to do this, we must have a machine that emulates human cognitive processes in order to foster efficient human-machine collaboration; however, it is not our goal to unnecessarily impose human cognitive limits on the machine.

2.1 Baseline Cognitive Model

This project began with a baseline computational model that was the product of two earlier efforts. Initially, inspired by research concerning naturalistic decision making and particularly, the Recognition-Primed Decision model (Klein, 1997), a conceptual framework was developed that provided the basis for modeling the cognitive processes whereby humans interpret events within naturalistic settings. In a subsequent project, this conceptual framework was instantiated as a computational model and expanded to provide mechanisms to account for a variety of organic influences on cognitive processes (e.g., arousal, stress, etc.). This instantiation provided the baseline, or starting point, for the current project.

As illustrated in Figure 1, the baseline model utilized three components. One component, Situational Knowledge, comprised a collection of situations, schema, themes, storylines, etc., pertinent to a given application. For example, in the initial security application, situations consisted of different tactics. In subsequent applications, situations have involved alternative interpretations of events and courses of action.

The second component, Associative Knowledge, contained cues representing various stimuli present within the environment. Associative knowledge was modeled using an associative network wherein each cue, often referred to as a concept, was represented by a separate node in the network and links between nodes indicated associative relationships between cues. With this design, when a cue was activated in response to environmental events, depending on the presence and strength of associative relationships, activation could spread to other associated cues.

The third component, Pattern or Situation Recognition, enabled the recognition of situations within Situational Knowledge in response to patterns of activation in Associative Knowledge. The baseline model employed an extremely simple approach in which a template was assigned to each situation in Situation Knowledge. This "fuzzy" template consisted of a set of cues that, if activated more or less in combination, would lead to situation recognition.

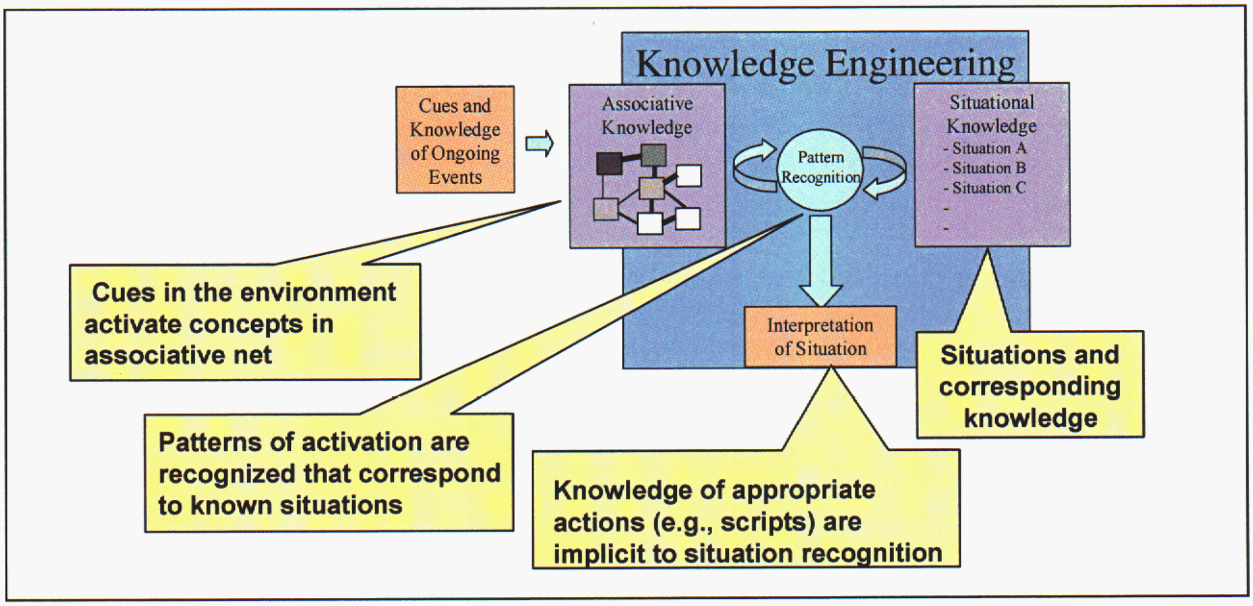

Figure 1. Baseline Cognitive Model.

The mechanisms underlying situation recognition were modeled using an oscillating systems approach. Specifically, each node in the associative network forming Associative Knowledge was represented by an oscillator. An oscillator consisted of a collection of individual neural units that responded in tandem. When a node was activated either by perceptual processes and/or spreading activation from an associated node, a subset of its neural units would activate with the number of activated units proportionate to the magnitude of stimulation from perceptual processes and/or spreading activation. If the number of individual units activated was sufficient, the node would begin to oscillate. This means that once activation of individual units was complete, another cycle of synchronous activation would immediately commence. Oscillations would continue in this manner until the external source of activation was removed, and residual activation dissipated. For a given node, amplitude reflected the number of neural units activated during an oscillatory cycle and frequency the duration of oscillatory cycles. Frequency and amplitude varied dynamically in response to a variety of factors (e.g., homeostatic state of individual neural units, generalized arousal, properties of momentum and energy dissipation, etc.).

The baseline model was instantiated using the commercial simulation MicroSaint and a demonstration problem involving a decision concerning the use of deadly force. For this demonstration, perceptual processes were scripted and there were no actions simulated. The capabilities demonstrated with the baseline model were rudimentary, at best. There was a general framework for representing knowledge and decision processes and underlying mechanisms consistent with an oscillating systems approach. However, vital components were unspecified (e.g., episodic memory), certain computational algorithms were unrealistically simplified, and the software framework did not provide a practical basis for system integration. Elaboration of the baseline model through the current project addressed these shortcomings significantly progressing capabilities toward their practical application.

2.2 Elaboration of Baseline Cognitive Model

There were four objectives in elaborating the design of the baseline cognitive model. These objectives are described separately in the following sections.

2.2.1 Formalization of the Design Process

While creation of the baseline cognitive model involved an extensive review of the relevant psychological and neurophysiological literature, the baseline design emerged from an informal process (Forsythe, 2001). One objective was to undertake a more systematic process that would provide thorough documentation of the design of the cognitive model, including the underlying design rationale.

In reformulating the design of the cognitive model, the first step involved reviewing the materials contributing to the baseline design, as well as other relevant materials. These reviews provided the basis for writing a set of specifications regarding the functional characteristics of the cognitive model. For example, Klimesch (1996) has provided evidence that the time required to perform tasks reliant on semantic (i.e., associative) memory positively co-varies with the dominant frequency in the alpha bandwidth. This finding was expressed in the following specification:

> "The rate of information processing for semantic tasks should be a function of the dominant frequency in the alpha bandwidth such that faster frequencies are associated with shorter task times."

The mechanisms by which the model addressed each specification were described in detail. The following description was provided for the above specification:

> "The level of activation for associative neural assemblies should vary in accordance with the salience of perceptual stimuli, emotional associativity, and potentially, other factors. As the level of activation in associative neural assemblies increases, momentum should develop with there being a corresponding increase in the frequency of oscillations. Thus, increased associative activation should lead to faster pattern recognition, and consequently, shorter task times."

In addition, for each specification, a substantiating citation, a brief plan for validating that the model satisfied the specification, and notes concerning constraints, qualifications and relationships to other specifications were provided. A total of 104 specifications of this nature were developed and incorporated into the revised cognitive model design. It should be noted that in collecting source materials, emphasis was placed on psychological and neurophysiological literature from humans that addressed cognitive processes associated with situation recognition as studied in various experimental research domains (e.g., narrative comprehension, naturalistic decision making, out-of-context paradigms, etc.), including related memory, emotional and perceptual processes.

2.2.2 Revised Cognitive Modeling Framework

The formalized design process resulted in extension, revision, and elaboration of the baseline cognitive model. A conceptual depiction of the extended model appears in Figure 2.

Perceptual processes provide the interface between external data sources and cognitive processes. Data sources are application-dependent and may include environmental sensors, database contents, system variables/states, etc. The goal has been to minimally constrain potential application of the model by allowing input to derive from a wide variety of sources.

In some cases, data may be preprocessed allowing it to be transmitted directly as input to post-perceptual processes (e.g., database contents). Otherwise, a framework has been developed wherein perceptual algorithms convert raw data into either discrete or continuous values consistent with the requirements of post-perceptual processes. The output of perceptual processes feeds into Associative Knowledge to activate concepts (i.e., cues) represented as nodes in an associative network. Separate perceptual algorithms correspond to specific cues within Associative Knowledge. These algorithms operate in parallel. The product of perceptual algorithms may vary in magnitude proportionate to the environmental stimulus (i.e., more salient stimuli may produce a stronger signal) with there being differential activation of corresponding nodes within Associative Knowledge. Additionally, perceptual algorithms may recognize multiple instances of a given cue (i.e., concept) that are each represented within a World Model as separate entities.

These developments represent a significant advance with regard to the baseline cognitive model. Previously, there was no framework for the model to receive input from external sources. Likewise, there were no mechanisms for data conversion consistent with perceptual processes. However, it should be noted that the framework described here is only an interim solution and several significant shortcomings are recognized. For example, it is not clear how perceptual processes may be affected by top-down influences. Similarly, mechanisms for the synthesis of separate data sources and subsequent construction of a world model are only preliminary. Further elaboration of perceptual components of the cognitive modeling framework is considered an important research and development topic with the eventual goal being an integrated framework in which perception emerges from the activities of low-, medium- and high-level processes, including high-level cognition.

Associative Knowledge functions similarly to the baseline cognitive model. However, there has been one significant modification. The baseline cognitive model represented each node in the associative network as a separate oscillator comprising a collection of distinct neural units. It was concluded that granularity at the level of neural units introduced unnecessary computational overhead and equivalent outcomes could be attained using a model in which individual oscillators served as the lowest level of granularity.

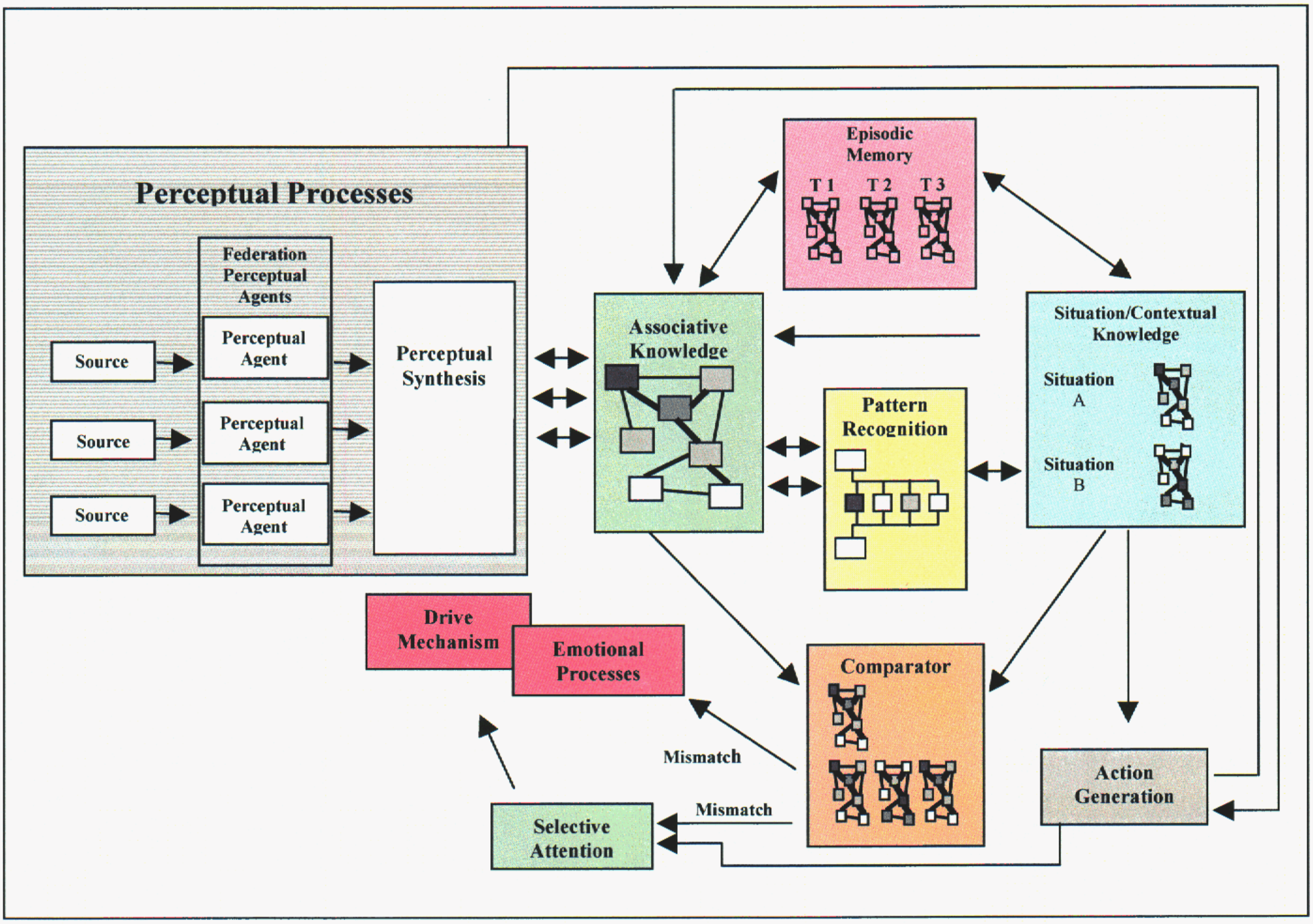

Figure 2. Conceptual Depiction of Cognitive Modeling Framework.

The baseline model used the term "Episodic Memory" to refer to the component of the current architecture that is now "Situational/Contextual Knowledge." As shown in Figure 2, Episodic Memory is now represented as a separate component. The intent was to distinguish schematic representations of events (e.g., party schema) from representations of specific episodes. From a functional perspective, this enables the model to interpret events on the basis of distinct schema when cues correspond to a known pattern. However, when there is insufficient correspondence between cues and known patterns, a model may recall the most similar past experience. While this design achieves the desired functionality, it should be noted that from a neurophysiological perspective, this is believed to be an artificial division. In fact, it is unclear whether schematic representations can be treated as distinct from either associative or episodic memory and a truer representation may be one in which schematic representations involve the convergence of associative and episodic memory.

The rudimentary template matching algorithm used in the baseline model for situation recognition was replaced by a more robust algorithm based on evidence accumulation. As with the baseline model, situations are evaluated in parallel during each cycle of the situation recognition module. The evaluation consists of determining the current level of "evidence" for each situation. Evidence is derived through multiple factors. For each situation, cues are identified that contribute either positive or negative evidence. A portion of the evidence accumulated during one cycle may carry over to the next. Also, there is a bias whereby once a situation has prompted action, evidence is sustained until the action is completed.

In addition to replacing the situation recognition algorithm, modifications have been made to allow multiple situations to be recognized simultaneously. Thus, within a multi-tasking environment, the model simulates conditions in which there is recognition of more than one ongoing situation. However, at present, mechanisms have not been incorporated to simulate the prioritization of situations and associated coordination of actions in accordance with situations.

Top-down activation was incorporated into the model such that each situation is attributed expectations. These expectations consist of cues within the associative network for which a sufficient likelihood exists that they will accompany the situation. For example, in a restaurant situation, the smell of food would be an expected cue. With the current architecture, once a situation has been recognized, activation is directed to cues identified as expectations of the situation that is proportionate to the level of expectation. Some cues may be highly probable and generate high levels of top-down activation, whereas the expectations associated with other cues may be relatively weak.

The effect of top-down activation is generally to lower thresholds enabling the activation of cues by less salient environmental stimuli, or in some cases, only by spreading activation from other cues. It should be noted that the current architecture only accommodates positive effects from top-down activation, and the incorporation of negative effects (i.e., negative priming) is an outstanding issue. Furthermore, the issue of negative effects is also unresolved for spreading activation that occurs within Associative Knowledge.

Within a given situation, cues may occur that are out-of-context for the situation. For example, a live elephant would be out-of-context in most restaurant settings. Within the current architecture, a Comparator detects and responds to stimuli that are out-of-context. As implemented, "out-of-context" is based on a cue not being identified as an expectation for a given situation. Consequently, an anomaly may arise under two circumstances. Within the context of a given situation, a cue(s) that is not an expectation for the situation may have sufficient activation to trigger the Comparator. Also, the overall activation for cues that are not expectations of a situation may similarly trigger an anomaly. Consequently, the Comparator responds to conditions in which there is either a single, or a few, highly salient out-of-context cues and conditions in which there are numerous out-of-context cues, although none are particularly remarkable. In either case, the magnitude of the Comparator response is proportionate to the activation of out of-context cues.

When the Comparator is triggered, there is an increase in arousal proportionate to the Comparator response. The Comparator elicits an arousal response through its influence on the pacemakers that produce the generalized rhythmic activation of oscillators underlying associative and situational knowledge. With associative knowledge, the frequency with which oscillators cycle is a function of the pulse rate for the pacemaker. With situational knowledge, the frequency with which evidence is updated is similarly a function of the pulse rate for the pacemaker. When triggered, the Comparator produces a temporary increase in the pulse rate for the pacemaker that gradually subsides in accordance with a specified decay rate. The effect is to temporarily increase the overall rate of information processing within the model.

When triggered, the Comparator also produces selective attention directed to the out-of-context cue(s). In the current model, selective attention takes the form of an orienting response. Specifically, the out-of-context cue(s) receives a temporary boost in its activation, while there is a simultaneous, generalized dampening in the activation of all other cues. The magnitude of these effects is proportionate to the Comparator response and diminishes at a specified rate of decay.

At present, the model does not accommodate selective attention effects on perceptual processes.

As previously noted, the Comparator is triggered when the activation for one or more cues that are not expectations for a currently recognized situation(s) exceeds some threshold. Alternative formulations for this threshold may be utilized singly or in combination. Consequently, the activation of out-of-context cues may be considered singly or in combination. Additionally, either absolute or relative levels of activation may be utilized. With relative values, the activation for out-of-context cues may be judged in relation to the activation of cues that are appropriate (i.e., in context) for the situation(s).

A given model is assigned a preferred threshold for triggering a Comparator response. By allowing the preferred threshold for individual models to vary, a population may be created in which individual models exhibit a differential responsiveness to out-of-context cues. Individual models assigned a low threshold will focus proportionately greater attention on out-of-context cues and, consequently, tend to more readily abandon situational interpretations of events. In contrast, models assigned a high threshold will only respond to the most salient out-of-context cues and, therefore, tend to maintain situational interpretations despite the contrary evidence present in out-of-context cues.

While an individual model is assigned a preferred threshold for its Comparator, the threshold is dynamic and varies in response to emotional processes. When activated, emotions assigned a positive valence (e.g., pleasure) will lead to a heightening of the Comparator threshold. Consequently, models will be less responsive to out-of-context cues and less likely to abandon situational interpretations of events. In contrast, emotions assigned a negative valence (e.g., dysphoria) will lower the Comparator threshold, increasing the model's responsiveness to out-of-context cues, lessening the likelihood of establishing and maintaining situational interpretations.

Currently, emotional processes incorporated into the model accomplish one primary functional objective. Through associations between emotional processes and other elements of knowledge, there is a heightened response to cues and events that are of particular significance. However, with subsequent versions of the model, emotions will play a vital role in certain forms of learning.

Separate emotional processes are specified for a given model including pleasure, dysphoria, frustration-anger, disgust and fear. Each is represented as a separate oscillator. While it is recognized that in humans, emotional processes may be directly activated by certain perceptual stimuli (e.g., loud noise), activation of emotions in the current model occurs as a product of cognitive processes. Particularly, elements within either associative or situational knowledge may be assigned an association with one of the emotional processes (e.g., the concept "snake" may be associated with fear). Emotional processes are activated in response to activation of concepts or situations for which an association has been specified, with the level of activation proportionate to the activation of the concept or situation.

With concepts, emotional processes produce a heightened activation of the concept triggering the emotional response. At the same time, the activation for all other concepts is dampened. As a result, information processing is focused on the eliciting stimulus to the deference of all other stimuli. In the absence of habituation, continued exposure to the eliciting stimulus produces a sustained emotional response. However, once the eliciting stimulus is removed, the emotional response subsides.

For situations, emotional processes function similarly to concepts. There is heightened activation of the situation triggering the emotional response while the activation for other situations is dampened. The magnitude of the emotional response is a function of the magnitude of activation for the situation that is derived through the situation recognition process and, therefore, is a function of the activation of associated concepts and their corresponding weights. Diminished activation for a situation lessens the magnitude of the emotional response. Likewise, the emotional response persists until the level of activation for the situation is no longer sufficient for recognition of the situation, after which the emotional response subsides.

Currently, conditions triggering emotions are specified in creating the model. The model may be extended so that emotions are triggered by generalized conditions allowing associations to be formed between emotional processes and elements of knowledge. Generalized conditions may take a form somewhat akin to "drives." For instance, at the most basic level, there may be conditions associated with material sustenance (e.g., hunger, thirst). When these needs are satisfied, the emotional process representing pleasure would be activated. The emotional response may then prompt associations to be formed or strengthened between the emotional process and the cues and situations activated either immediately preceding or in conjunction with activation of the emotional process. At a somewhat higher level of abstraction, emotional processes may be similarly activated in response to either the attainment of goals or blocked goal attainment. Likewise, physical pain and exposure to too little or too much stimulation may serve as generalized conditions for triggering emotional processes.

2.2.4 Conceptual Design for Episodic Memory

The overall objective for this project has been to develop a capability for cognitive models to store a meaningful representation of their experiences and apply knowledge of their experiences in a beneficial manner. The extension and elaboration of the baseline cognitive model described in previous sections were a prerequisite to attaining the capability for a meaningful representation of experience.

In creating a conceptual design, relevant literature was reviewed to identify a collection of functional characteristics of human episodic memory that a computational model should exhibit to be considered a realistic representation of human episodic memory. Table 1 lists the characteristics that were chosen as vital to attain a realistic model of episodic memory.

Table 1. Functional Characteristic Considered Vital to a Realistic Computational Representation of Human Episodic Memory.

Characteristic	Instantiation	Source(s)
A record of ongoing experiences should be stored that includes both contents (i.e., specific details of event) and context.	As model operates, an episodic memory store consisting of a series of traces is continually amended. Each trace written to episodic memory consists of values representing activation of associative and situational knowledge, as well as emotional processes.	Nyberg and Tulving (1996); Wheeler et al. (1997)); Dosher and Rosedale (1991)
The representation of episodic memory should be functionally distinct from other memory representations.	Episodic memory modeled as a component that is distinct from other memory representations (e.g., associative and situational knowledge), however its operations are thoroughly integrated with these other memory representations (e.g., elements from associative knowledge are not explicitly represented in episodic memory, but their activation provides the basis for episodic recall of events).	Wheeler et al. (1997); Krause et al. (1998); Wiggs et al. (1998)
Knowledge of experience should be stored on the basis of prototypical schema, as opposed to specific details of event, except for cases where actual events deviate from prototypes.	Recognition of a situation (i.e., schema) serves as a trigger for writing a trace to episodic memory. Independent of situation recognition, significant events (activation of associative elements or emotional processes) may trigger an episodic trace enabling recall of atypical events.	Trafmow and Wyer (1993)
Recognition performance based on episodic memory should be a function of dispersed activation in the theta bandwidth.	Recognition involves determining if an event with specific characteristics occurs within episodic memory. Traces for events may be written to episodic memory without situation recognition and similarly recalled based on characteristics of the event captured within the episodic trace. While computationally represented as separate processes, there is no functional distinction, with both processes utilizing the same theta-band oscillator. However, whereas the correspondence to a known situation generates the level of theta-band activation for situation recognition, the extent to which the probe matches an episodic trace produces the level of activation during recall of specific events. Consequently, theta-band activation should be greatest for stimuli that match a trace(s) in episodic memory.	Klimesch et al. (1994)
Increasing demands by manipulating the relative complexity of stimulus presentations or the presence of distracters should lead to increased activation in the theta bandwidth.	With increased stimulus complexity or distracters, there is a greater incidence in which partial matches occur between stimuli and traces within episodic memory. The overall theta-band activation is a function of the combined activation across episodic traces. Consequently, theta-band activation should be greatest for conditions in which the number of traces for which there exists partial matches is greatest.	Klimesch et al. (1994)

There should be a gradual increase in activation prior to and accompanying recollection so that reaction time co-varies with the latency in the peak response amplitude.	Episodic memory recall utilizes the same evidence accumulation algorithm as situation recognition, however recall cues dynamically fill the terms in the equation and activation is based on the activation levels written to episodic traces. Therefore, during recall, accumulating evidence corresponds to increasing activation until evidence reaches the threshold for recognition.	Wilding (2000); Rugg and Coles (1995)
The latency of peak activation should be a function of the difficulty of a recognition task.	With difficult tasks, evidence builds more slowly due to cue ambiguity. Consequently, activation accumulates more slowly with a greater latency prior to attaining peak latency.	Coles et al. (1995); Kok (1990)
Mechanisms should exist for representing and retrieving temporal properties of experience.	Each episodic trace includes a timestamp that, where appropriate, provides a precise record of time. Where a realistic representation is desired, including biases in temporal judgments, there are temporal properties (e.g., sequence and number of traces) inherent to the episodic record of experience, including cognitive processes giving rise to biases in temporal judgments (e.g., activation of emotional processes).	Zwaan and Radvansky (1998); Allen et al. (2000); Shum (1998); Bagaley (2000); Tulving and Kroll (1995)
Mechanisms should exist for representing spatial properties of experience.	Place information represented within associative knowledge is written to episodic traces. The current cognitive framework does not include a representation of spatial knowledge. Episodic memory may be expanded to include mechanisms to store referents to a spatial knowledge representation.	Zwaan and Radvansky (1998); Allen et al. (2000); Taylor et al. (1999); Millis (1994); Bagaley (2000)
Mechanisms should exist whereby the causal structure of events provides a basis for segregating memory of experiences.	Causal attributes of experience are captured within situational knowledge. Episodic traces contain values representing the activation of elements of situational knowledge, and recognition of situations provides a trigger for writing an episodic trace.	Zwaan and Radvansky (1998); Kerstholt and Jackson (1998)
Mechanisms should exist whereby the motivational and intentional structure of events provides a basis for segregating memory of experiences.	Intention (i.e., goals) and motivations (i.e., emotional associations) are captured within situational knowledge.	Zwaan and Radvansky (1998)
Mechanisms should exist whereby emotional aspects of events provide a basis for segregating memory of experiences.	Episodic traces contain values representing the activation of emotional processes. Activation of emotional processes occurs in response to activation of elements of associative or situational knowledge which serve as a basis for memory segregation.	Zwaan and Radvansky (1998)
Mechanisms should exist to represent person- and object- related facets of experience and segregate experience accordingly.	Both person and object facets of experience would be captured within episodic traces through the representation of the activation of elements of associative knowledge.	Zwaan and Radvansky (1998)

There should be mechanisms whereby certain perceptual facets of experience are represented in episodic memory and recall leads to activation of equivalent perceptual processes.	The current cognitive framework provides a minimal representation of perceptual processes and associated memory mechanisms. While no benefit is seen for storing the activation of current perceptual processes in episodic memory, episodic traces may be expanded to include referents to appropriate perceptual processes.	Fletcher et al. (1996); Uttl and Graf (1996)
There should be a mechanism enabling experiences to be recalled on the basis of combinations of cues that include time, place, objects, entities, etc.	Episodic memory may be searched on the basis of concepts, situations or emotions, singly or in combination, to identify instances of their occurrence. This retrieval mechanism may be applied to any feature appearing within the episodic trace. Once identified, an experience (i.e., constituent traces) may be replayed allowing the experience to be recalled.	Wheeler et al. (1997); Moll (1997); Baguley (2001)
Recall of episodic memory should influence ongoing semantic memory processes.	The cognitive framework utilizes a single representation of associative knowledge. Consequently, the same elements of associative knowledge are activated by ongoing events and episodic recall. Therefore, the opportunity exists for priming and related influences as a result of episodic recall.	Klein et al. (1996); Dosher and Rosedale (1991)
Associative knowledge should influence the recall of events.	During the actual experience, both spreading activation between related concepts and top-down activation may produce activation of elements of associative knowledge independent of bottom-up processing of events. This activation is written to episodic memory without any distinction with regard to whether activation was produced by bottom-up or other processes. Furthermore, during recall spreading activation and top-down influences will operate similarly. Finally, deliberative recall mechanisms may be implemented that rely on situational knowledge where episodic memory is incomplete or cannot be retrieved.	Sherman and Bessenoff (1999); Dosher and Rosedale (1993)
There should be greater recall for events, including actions, that were actually experienced than for those that were only observed.	Recall is a function of the activation (e.g., elements of associative knowledge) recorded during the event. Activation resulting from first-person experiences should generally be greater than that for third-person experiences due to the salience of perceptual cues and heightened activation produced through action initiation.	Zimmer et al. (2000); Wang (1999)
Retrieval failure may result from stimulus generalization (cues appear within multiple contexts).	In recall, episodic memory is scanned to identify instance with the greatest level of activation. Where there is an insufficient basis to discriminate experiences on the basis of activation (i.e., same level across numerous occurrences), this condition may be treated as a retrieval failure.	Bouton et al. (1999)
Retrieval failure may result from contextual factors (a given context presents competing cues).	Activation of elements of associative knowledge is differentially influenced by contextual factors (i.e., top-down activation). Consequently, recall for equivalent stimuli may differ as a result of the context in which the stimuli are presented.	Bouton et al. (1999)

Retrieval failure may occur due to forgetting (fading attraction with time).	Activation associated with episodic traces should spontaneously decay, unless refreshed through recall of experiences. Consequently, the more recent of two equivalent experiences would be recalled.	Bouton et al. (1999)
Recall from episodic memory should favor generalized schema, as opposed to detailed representations of objects, individuals, scenes, etc.	With the decay of memory across time, only elements written to episodic traces with the greatest levels of activation (i.e., the most salient features of the event) will be recalled. Mechanisms may be implemented such that situational knowledge is employed to construct recall of experiences that have largely faded from memory.	Sherman and Bessenloff (1999)
For items that are out-of-context relative to the ongoing situation, there should be greater recall than for items that are contextually appropriate.	Out-of-context cues trigger a response from the Comparator heightening the associated activation for the cue in the associative network. Consequently, as episodic memory fades, there should be relatively greater persistence for out-of-context cues.	Schmidt (1996); Rugg and Coles (1995)
For contextually appropriate cues, there should be greater recall of cues that are most typical of the context or situation.	Top-down activation should be proportionate to the typicality of cues relative to a situation or context leading to heightened activation of typical cues, as opposed to less typical cues. Consequently, there should be greater persistence of episodic memory for these cues.	Schmidt (1996)
Episodic memory should not be a passive recording of experience but, to some extent, the product of constructive processes.	Constructive processes are inherent to the cognitive framework through top-down processes that subsequently shape the contents of episodic memory.	Baguley (2000)

The baseline cognitive model, with the extensions discussed in preceding sections, provided the essential elements for creating a computational model of episodic memory. The primary objective was a capability to store and retrieve a meaningful representation of experiences. Therefore, associative and situational knowledge, including emotional associations, supplied the basic contents for filling episodic memory records.

The contents of episodic memory consist of a series of traces[1]. At certain instances, the activation of concepts in the associative network, situations and emotional processes are written to episodic memory. These values constitute a trace providing a representation of the overall state of the cognitive model at a specific point in time. Specifically, a trace contains: (1) a timestamp, (2) the identity and level of activation for all concepts in the associative network with a level of activation above a threshold, (3) the identity and level of activation for all situations with a level of activation sufficient for recognition and (4) the identity and level of activation for

[1] It is important to distinguish our use of the term "trace" from the mediationist meaning for trace (Watkins, 1990). Specifically, in this context "trace" is simply a term that is synonymous with "memories" or "memories of specific episodes." In no way do we make the assumption that there are separate and distinct physical entities that encode or encapsulate these memories. That is, in no way does the word "trace" make any statements about the knowledge representation method used in the Sandia cognitive framework. Further, with regards to the computational implementation of a trace in episodic memory, the information contained in the computational traces, again, is not assumed to be encapsulated in physical entities contained somewhere in the human brain.

any emotional processes that are activated. Whereas episodic memory should also contain a representation of aspects of experience associated with perceptual processes, spatial knowledge and action execution, these components are not currently implemented. However, the format of an episodic trace may be readily expanded to incorporate these dimensions.

Episodic memory does not involve a continuous record of experience but is discontinuous with traces written in response to triggering events. Currently, triggering events include: recognition of a situation, pronounced activation of a concept in the associative network or pronounced activation of emotional processes. For the two latter triggers, the intent is to capture salient events (e.g., a highly salient stimulus appears), as opposed to an ongoing history of experience.

Episodic memory retrieval may occur through two mechanisms. The first involves an intentional recall process whereby, given certain cues, the objective is to recall one or more past experiences. Here, retrieval takes a form not unlike typical search functions (e.g., SQL database search). A query may be specified using cues, situations or emotions, or any combination of cues, situations and emotions. For example, a query may be formed using the cue "open flame grill" and situation "restaurant." Retrieval would consist of scanning episodic memory to identify traces involving this cue-situation combination. Multiple instances may exist, however the one or more instances with the greatest combined level(s) of activation would be retrieved.

In retrieving an episode, once the trace with the greatest activation is identified, an episode is recalled that may consist of a sequential series of traces. This involves partitioning the record by locating prominent situational transitions (i.e., traces in which there was a pronounced transition between situations). Thus, the recalled episode begins and ends with a prominent situational transition and includes all intervening episodic traces. At recall, the traces are played back with their constituent levels of activation for concepts in the associative network serving as input to the associative knowledge component of the model. Situation recognition and activation of emotional processes then occur as a spontaneous response to activation of concepts in associative knowledge. Finally, it should be noted that during recall interactions occur due to activation of the cognitive model in response to ongoing events and the cognitive model's knowledge structure (e.g., associative relationships between concepts in associative knowledge) including revisions to the knowledge structure that have occurred since the actual episode.

The second retrieval mechanism serves to supplement the situation recognition processes. Without episodic memory, a model can only interpret events with regard to generalized schema (e.g., restaurant). Episodic memory enables two additional functions. First, events may be interpreted on the basis of their similarity to some past experience (e.g., a specific car problem). In this case, episodic memory retrieval operates in parallel with recognition based on generalized schema. The mechanism may involve either an equivalent or the same evidence accumulation process. However, cues present within the ongoing situation provide the terms for calculating evidence. Specifically, episodic memory is queried using the recall mechanism described above and current cues as the search terms. The episode retrieved through this query is then assessed to establish its associated level of evidence. For this calculation, the level of activation in the episodic trace for each cue present in the current situation is summed, and activation levels for cues not present may be subtracted. Generally, for events where there are sufficient cues for recognition of one or more generalized schema, interpretation of the situation will be based on generalized schema. However, episodic memory retrieval offers an alternative means of interpreting events, particularly for ambiguous cases where there are insufficient cues for interpretation based on generalized schema.

The second function concerns the updating of situational knowledge. Given either a particularly salient episode or repeated exposure to equivalent episodes, the episode may be represented as a unique situation within situational knowledge. For instance, a highly emotional event for which there is an unusually high level of activation may be represented as a unique situation.

Finally, a decay function is applied to episodic memory so that as time passes, the level of activation associated with a given trace slowly diminishes. The effect is to favor retrieval of recent episodes where episodic memory contains multiple episodes that are essentially equivalent. Furthermore, complete decay operates as a compression mechanism. With time, all but the most salient episodes fade from memory; and, in the case of those episodes, knowledge of all but the most salient cues similarly fades.

3. Implementation Studies

3.1 Project Year 1: Building Surveillance Implementation

This implementation explored the feasibility of a machine-based episodic memory structured on the basis of themes. Furthermore, an approach was utilized in which themes were statistically derived using data generated through simulation.

For this implementation, simulations were created in which eight robotic vehicles systematically searched a building to locate a smoke source. Based on their sensors, communications and data processing capabilities, as they progress through the scenario, different concepts in their semantic networks were activated (see Figure 3a). The result was a time series of patterns of semantic activation. This time series was statistically analyzed to identify recurrent schema (e.g., progressing down a hallway following a smoke gradient). This is illustrated in Figure 3b. Endowed with knowledge of these schema, stories could be constructed that are based on the sequence of schema experienced during a given event (see Figure 3c). Additionally, subsequent analysis allowed identification of recurrent sequences of schema (i.e., themes or storylines).

3.1.1 Software Implementation

The cognitive architecture is being implemented in C++, building on Sandia's Umbra framework for efficient, highly-modular simulation (Gottlieb et al., 2001). Components in the Cognitive Architecture are implemented as C++ objects known as Umbra modules. Data interfaces and functional interfaces needed by the components determine base classes for the different component types. A Tcl shell/script interface is used to construct particular subclass instances, configure them, and connect their interfaces at runtime. The modular approach makes it easy to try out different models of a component, for example to replace a conventional pattern recognizer with an artificial neural network.

The Semantic Knowledge (SK) network, the most complex component of the Cognitive Architecture currently implemented, serves as an example. There are two main parts to the SK network implementation: a collection of concept nodes and a main SK node that manages this collection. The manager node handles the creation of concept nodes, their connection into an associative network, and ensuring their update functions are called when necessary.

Each concept node in the associative network is implemented as an Umbra module with a secondary module summing its spreading-activation inputs. Concept nodes have a set of inputs (pacemaker pulse, activation threshold, sensory/perceptual stimulation, spreading activation, intrinsic stimulation, psychogenic agents, drive associativity, etc.) that parameterize their activation. The output data interface of the SK network is a (software) bus of the current activation levels of the concept nodes. This bus is connected to the Pattern Recognizer, Episodic Memory, and Comparator by the simulation start-up script. When only a single module determines a concept node input, they can be directly connected at start up. Otherwise, a functional interface provided by the manager node is used. For example, the Pattern Recognizer and the Situation/Contextual Knowledge components both need to adjust the activation thresholds of concept nodes.

Additionally, the SK manager node's functional interface has other uses, such as to enable the World Model to establish or break bindings to concept nodes. Finally, module input values and connections between module inputs and outputs can be dynamically changed interactively or through scripts during execution.

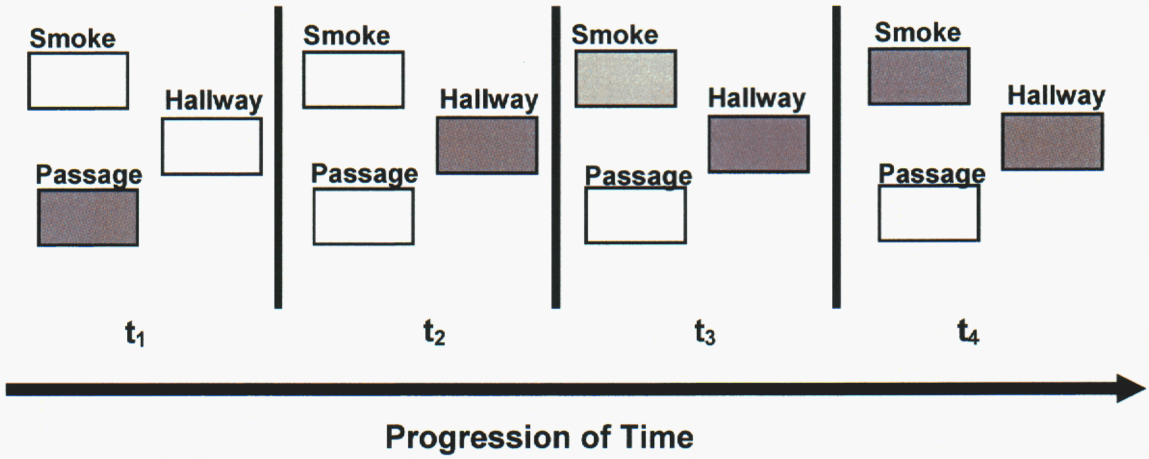

Figure 3a. Time Series of Patterns of Semantic Activation.

R1 Goes Down Hallway with Increasing Smoke

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
[illegible]	[illegible]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R1 Smoke	0	0	0	0	1	2	3	4	4	3	3	3	4	4	4	5
R1 Passage	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0
R1 Hallway	0	0	1	1	1	1	1	0	1	1	1	1	0	0	0	1
R1 Intersection	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
R2 Alarm	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R2 Smoke	0	0	0	0	0	1	3	5	7	8	8	8	8	7	8	8
R2 Passage	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
R2 Hallway	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	0
R2 Intersection	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1
R1–R2 Direction	1	1	0	1	1	1	1	1	1	1	0	0	0	1	0	0
R1-R2 Separation	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	4

Figure 3b. Derivation of Schema Based on Recurrent Patterns of Semantic Activation.

1. Entered building
2. Searched for smoke, found no smoke
3. Selected path, passage into hallway
4. Followed path (search smoke)
5. Detected smoke
6. Followed path (smoke gradient), reached intersection
7. Sampled paths, found path with more smoke
8. Followed path (smoke gradient), reached intersection
9. Alerted (destination)
10. Followed path (destination)

Figure 3c. Story Generated Based on Sequential Ordering of Schema at the Conclusion of Simulation Run.

A total of 20 simulation runs, each involving eight robots, were conducted within the framework of Umbra (Gottlieb et al., 2002). Umbra enables the simulation of multiple autonomous agents with a variety of physical phenomena such as RF (radio frequency) communications, interactions with solid objects (e.g., collisions), ultrasound communication, IR (infrared) detection of objects, vehicle physics, terrain descriptions, and other phenomena. All of these physical attributes can be simulated simultaneously with a graphical visualization that allows the monitoring of the vehicles' performance over the terrain.

The vehicles are models of actual hardware. Each vehicle contains four IR sensors for detecting objects between 0.15m and 0.46m on its four sides (see Figure 4). The vehicles also contain RF communication devices to be able to converse with other vehicles within a 30m line-of-sight (LOS) or roughly 10m through walls. They also have ultrasound capability to measure the distance between them provided they are within 10m of each other and in LOS. The vehicle physics models are simple and proved adequate on a smooth surface. The building model was generated as a CAD model and contains several connected hallways as well as a multitude of variable size rooms. The control algorithms for the vehicles allowed them to avoid contact with walls and other vehicles. Beyond that, the control algorithms enable the collective to place a member at the maximum smoke concentration found in the building. Note that a strict mathematical model of this situation is intractable. This is due to both the discrete event-based nature of the communications as well as the dynamic physics models with very complicated interactions between the vehicles and obstacles. Thus, the simulation shows stability in a qualitative rather than strictly mathematical fashion.

The restriction that vehicles cannot move through walls, doors, or each other essentially ensures they remain inside the building. This is accomplished via rules that use the IR sensors to follow walls down a hallway. This enables the vehicles to move throughout the building, though not necessarily in any prescribed fashion. Further restrictions on the vehicles involve the maintenance of a continuous RF communication network requiring that vehicles stay within 30m of each other or less if LOS is lost (i.e., they may have to stay at a wall junction to maintain LOS).

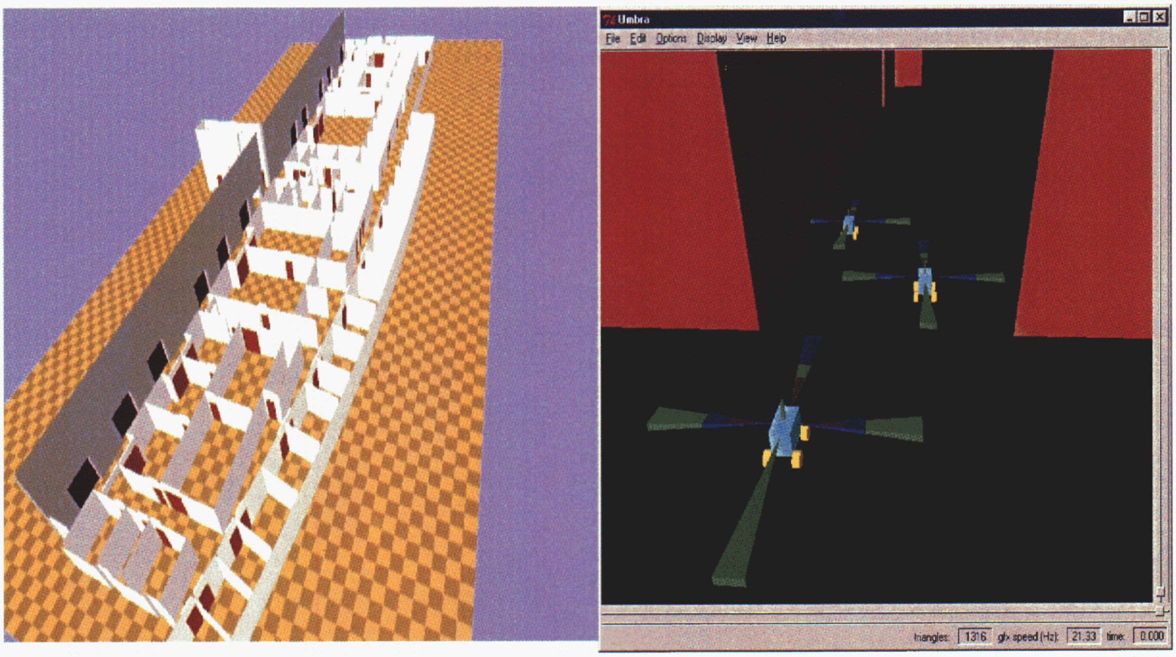

Figure 4. Detailed Simulation of Multiple Vehicles Navigating a Building.

For the eight simulation runs, the source of smoke varied from simulation to simulation so as to induce different behavior across runs. For these simulations, the initial location/status of the eight robots was constant across runs. Each robot possessed a cognitive model consisting of an associative network in which nodes received input concerning the numeric values for each sensor and actuator on the robot. It should be noted that since the intent was to statistically derive the contents of situational knowledge, cognitive models did not contain situational knowledge or situation recognition modules. Additionally, there was not spreading activation between nodes in the associative network.

The level of activation of nodes in the associative network of robots was recorded and provided the data set used in subsequent statistical analysis. A complex, multi-dimensional data vector (with binary and continuous dimensions) was used to define the status of each robot at any point in time during a simulation. The status of each robot was sampled once per second over the duration of each 300-second simulation run. Thus, there are 300 observations per robot per simulation run. Thus, the total data set consists of 48,000 observations, where each observation consists of the 15 dimensions listed in Appendix A. Figures 5 and 6 provide *plan views* of robot paths for two representative simulation runs. Note that for each simulation run, the initial robot positions are spread out in a line segment approximately at x=11 between y = -2 and y=28.

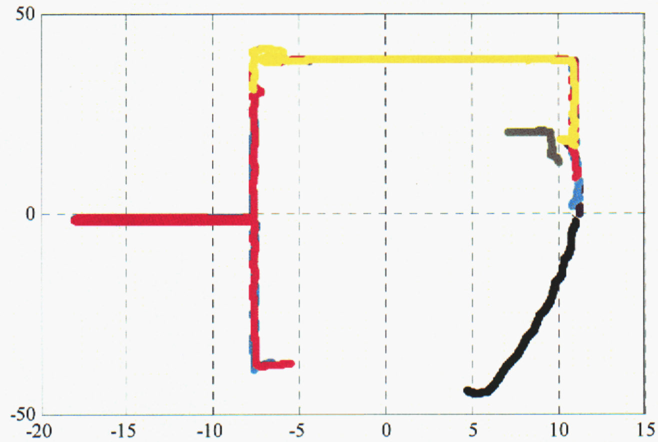

Figure 5. Color-Coded Robot Paths (Simulation #5).

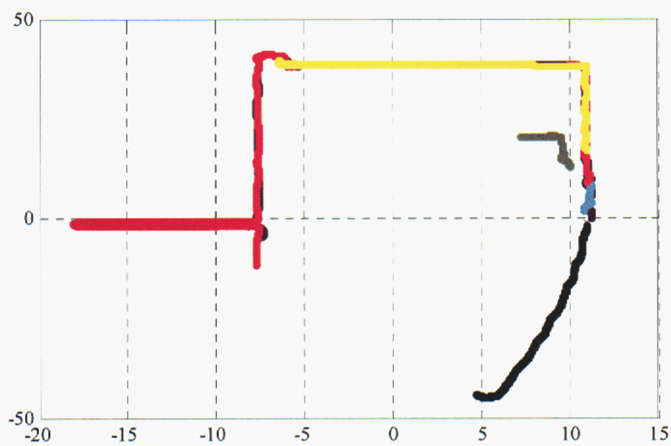

Figure 6. Color-Coded Robot Paths (Simulation #18).

The analysis consists of several distinct steps. First, using a representative training set, *cluster analysis* was used to group the collection of observations into subsets (or *clusters*). Interpretation of these clusters was facilitated via a *classification tree* model. All observations (over all simulations/robots) were then partitioned by the classification tree rules into interpretable robot states. At this point each observation has been mapped from the complex, multidimensional data vector into a discrete state-space (with relatively few states). This dimension reduction facilitates the analysis of temporal patterns exhibited by individual robots as well as the system of robots. In addition, it is relatively easy to study differences in behavior from robot to robot and across simulation runs.

3.1.1 Cluster Analysis

Cluster analysis is a form of *unsupervised learning* where the goal is to partition a collection of observations into subsets (or clusters) such that those observations within a cluster are more closely related to one another than observations assigned to different clusters (Kaufman and Rousseeuw, 1990). The nature of unsupervised learning is that there is no knowledge of the true data structure. Cluster analysis methods are simply algorithms. In this analysis, two clustering algorithms were used: K-means clustering and DIANA.

The K-means algorithm begins with guesses for the number of clusters and location of each cluster's multidimensional center. It then iterates the next two steps until convergence.

1. For each observation identify the closest (in Euclidean distance) cluster center.
2. Replace each cluster center with the average of all points that are closest to it.

Convergence is declared when the cluster assignments do not change. In practice, one starts the K-means algorithm a number of times, each time with a different specification for the number of clusters. At convergence for each case, the total within-cluster variability is used as a measure to select the number of clusters. The goal is to obtain a partitioning that involves relatively few clusters such that the level of within-cluster variability is acceptably small. Another goal here is to develop a set of clusters such that the number of observations per cluster is not too small.

DIANA is a clustering algorithm that, unlike the K-means algorithm, is hierarchical in nature. That is, clusters at each level of the hierarchy are defined by combining clusters at the next lowest level. There are two general approaches for hierarchical clustering: *agglomerative* (or bottom-up) and *divisive* (or top-down). Agglomerative approaches start with each observation defining a singleton cluster. Divisive approaches start with a single cluster containing all observations. DIANA is a divisive clustering algorithm where at each stage the cluster with the largest *diameter* is selected to be partitioned. The diameter of a cluster is the largest dissimilarity between any two of its observations. Here, Euclidean distance was used as the measure of dissimilarity. To partition the selected cluster, the algorithm first looks for the observation that has the largest average dissimilarity to the other observations within the selected cluster. This observation becomes the initial element (observation) of a new cluster. In subsequent steps, the algorithm reassigns observations that are closer to the new cluster than to the remaining elements of the selected cluster. The result is a partitioning of the selected cluster into two new clusters. The goal is to identify a position in the hierarchy with relatively few homogeneous clusters such that each cluster has a reasonable number of observations.

There are many other variants of cluster analysis that could have been used here. Some of these other algorithms might have partitioned the data in more useful ways than the methods that were used. Nevertheless, the scope of this analysis was limited to the two variants that were described.

3.1.2 Classification Trees

Classification tree modeling is a form of *supervised learning* where the objective is to partition the predictor variable space into regions that are homogeneous with respect to *known* classifications. The method is divisive as it starts with all of the observations in a single node and proceeds through a series of binary splits to partition the observations. Each binary split separates the observations comprising a node into two daughter nodes via comparison of a single predictor variable with a threshold value. For each split, the particular predictor variable and associated threshold value are selected in order to achieve relatively homogeneous daughter nodes in terms of cluster assignment. Ultimately, a tree structure results where the terminal nodes of the tree contain observations that are predominately from a single class. The pathway to each terminal node consists of satisfying a series of logical comparisons of one or more predictor variables with various threshold values. The process requires *a priori* assignment of observations to clusters. Here, the *known* classifications are the clusters that were determined from cluster analysis. The predictor variables are described in Appendix A. The main purpose of using classification trees is to enhance the *interpretability* of the clusters that were identified by the cluster analysis.

3.1.3 Data Analysis – Cluster Analysis and Classification Tree Modeling

Recall that the total data set consisted of 48,000 observations. It was not possible to perform the cluster analysis and classification tree modeling with such a large data set due to computational limitations. Therefore, the cluster analysis and classification tree modeling was based on a representative training set consisting of 800 observations. The training set was obtained by randomly selecting ten observations per robot per each of the first ten simulation runs. Thus, we have representation across all robots and simulation runs.

For both the K-means and DIANA algorithms a range of values from one to ten was considered for the number of clusters. In the case of K-means, five clusters appeared to provide an adequate partitioning of the training set. In the case of DIANA, six clusters provided a reasonable partitioning of the training set.

A classification tree analysis (using an SPLUS implementation) was performed given the sets of cluster associations developed by the K-means and DIANA algorithms. Figures 7 and 8 illustrate an interpretation of the tree structures that result in five terminal nodes (states) in the case of the K-means algorithm and six states in the case of the DIANA algorithm. The tree structures can be interpreted as follows.

In Figure 7, the primary partitioning of data was with regard to dimension-15 (an indicator of whether the robot was or was not stopped). In particular, observations with a value of less than 0.8 for dimension-15 were passed to the left side of the tree and to the right side otherwise. State-5 is associated with a "moving robot" ***and*** a small value for dimension-4. That is, state-5 relates to a robot moving slowly in the x-direction. State-3 is associated with a robot that is moving relatively quickly in the x-direction. State-1, state-2, and state-4 are associated with robots that have stopped (or nearly have stopped). The difference between state-1 and {state-2, state-4} is due to dimension-8 (an indicator of how close the robot's current smoke level is to its previous maximum smoke level). Thus, state-1 pertains to robots that have stopped at a position where the smoke level is not close to the maximum smoke level that had previously been experienced by that robot. The difference between state-2 and state-4 is the presence/absence of an RF_Ping (dimension-14). For example, state-2 is associated with robots that have stopped ***and*** are at a position where the smoke level is close to the maximum ***and*** are not pinging.

In Figure 8, the dimensions that lead to the definition of the state space are: dimension-13 (RF_Hear_Beacon), dimension-6 (current smoke level), dimension-4 (level of movement in x-direction), and dimension-5 (level of movement in y-direction). For example, state-1 is associated with robots that are not currently hearing a strong beacon signal ***and*** are detecting relatively low levels of smoke. Also, for example, state-4 is associated with robots that are hearing a strong beacon signal ***and*** moving quickly in both the x- and y-directions.

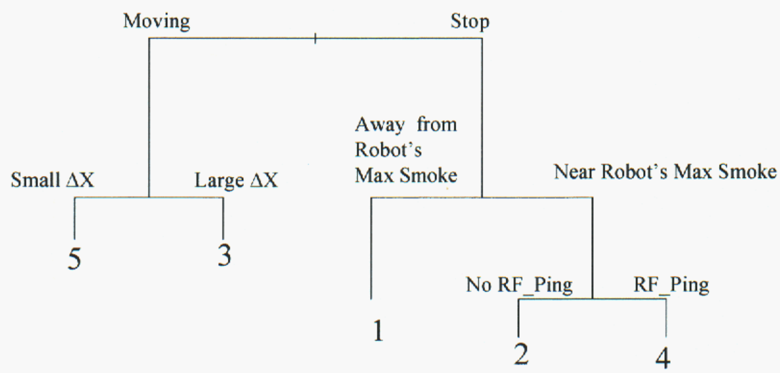

Figure 7. Classification Tree Derived From K-Means Clustering.

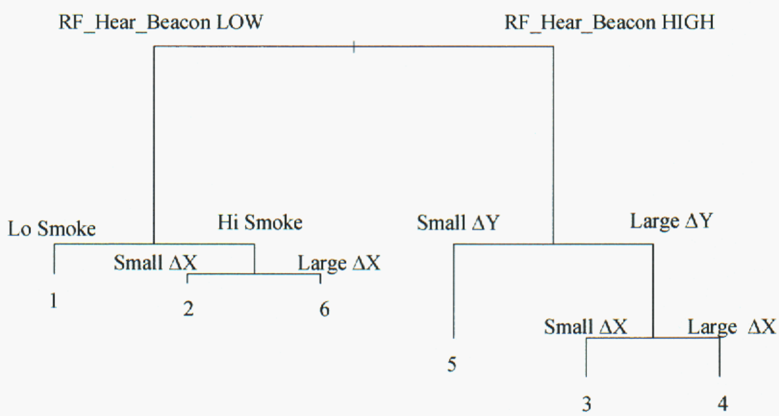

Figure 8. Classification Tree Derived From DIANA Clustering.

3.1.4 Data Analysis – Analysis of State Spaces

The classification trees were developed using a training set of 800 observations. The partitioning rules associated with these trees were applied to the complete set of 48,000 observations. Thus, each of the 48,000 observations were assigned to a particular state (states 1-5 in the case of trees developed from the K-means clustering and states 1-6 in the case of the DIANA clustering). Figures 9-14 display summary information concerning the relative frequency of these states (overall: Figures 9 and 10, by robot: Figures 11 and 13, and by simulation run: Figures 11 and 14). Figures 15 and 16 display the frequencies of transitions between states for each robot. Figures 15 and 16 each contain an 8 x 5 array of bar charts. Each bar chart provides the frequency (on a logarithmic scale) of state transitions ($S_t \rightarrow S_{t+1}$) for each robot over all simulation runs. For example, in Figure 15 consider the bar chart pertaining to robot-1 and S_t = 2. The bar chart clearly shows that in most cases robot-1 remains in state-2 during the next epoch. That is in about 95 percent of all instances with robot-1, $S_{t+1} = 2$ given that $S_t = 2$. In other instances $S_{t+1} = 4$. Thus, together with subplot in Figure 9 representing robot-1, the interpretation is that this robot is always *stopped* and occasionally *pinging*.

The different state space representations (K-means and DIANA) can be integrated to investigate the behavior of individual robots. The sets of figures pertaining to individual robots ({Figures 10 and 14} and {Figures 13 and 16}) are used to facilitate this investigation. Based on the analysis associated with K-means clustering, robot-1 clearly stands out by virtue of the fact that it resides exclusively in state-K2 and state-K4 (the *K*-prefix denotes a state associated with the K-means tree). Based on the analysis associated with DIANA clustering, robot-1 is also found to exhibit unusual behavior as it resides entirely in state D5 (the *D*-prefix denotes a state associated with the DIANA tree). Additional comparisons of the patterns in Figures 10 and 15 and Figures 14 and 16 indicate five classes of robots with regard to their behavior: {robot-1}, {robot-7}, {robot-8}, {robot-2, robot-3}, and {robot-4, robot-5, robot-6}. [One might argue that {robot-7} belongs with {robot-4, robot-5, robot-6}.]

One might summarize the different robot behaviors as follows. Robot-1 is the least mobile robot. It is always *stopped*, always hears a strong beacon signal, and is occasionally *pinging*. Robots-{2,3} are usually stopped or moving slowly, near high levels of smoke, and are not hearing a strong beacon signal (states K1, K2, and D2). Perhaps these robots lead the way in exploring for the source of smoke. Robots-{4,5,6} spend their time in a variety of states (most frequently D2, D3, and K5) Robot-7 behaves similarly to robots-{4,5,6}. However, robot-7 spends a larger proportion of time in state D2, which can be viewed as a *terminal* state. That is, once a robot enters this state, it is unlikely to leave it (see second column of Figure 12). Note that entry to D2 is exclusively through D6. Robot-8 is somewhat similar to robots-{4,5,6} and robot-7. A notable difference is that robot-8 did not transition from K2 to K3 (see Figure 11) giving rise to significantly fewer visits to K3 than robots-{4,5,6}.

Finally, one might want to investigate the system (joint) behavior of the robots within a simulation. Two interesting time points within the simulations are at the beginning and at the end. Figures 17 and 18 illustrate the distributions of robot states at the beginning (epoch-1) and end (epoch-300) of the simulations for each state space representation. The distributions span all robots and all simulations.

In terms of the K-Means state space, the robots generally start in K5 (an exception is robot-1 who starts in K2). At the end of the simulations, most robots have stopped (K2 and K1). In terms of the DIANA-state space, the robots begin in D3 with the exception of robot-1 who begins in D5. At the end of the simulation, most of the robots are in D2 and to a lesser extent, D5. One could further analyze the ending states, by robot.

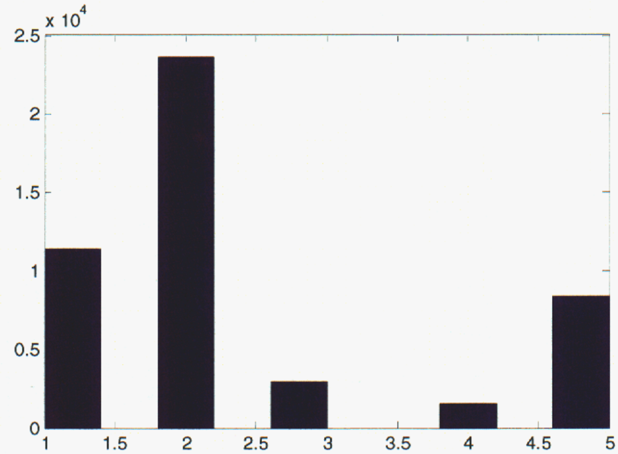

Figure 9. State Frequency: Using Classification Tree Developed From K-Means Clustering.

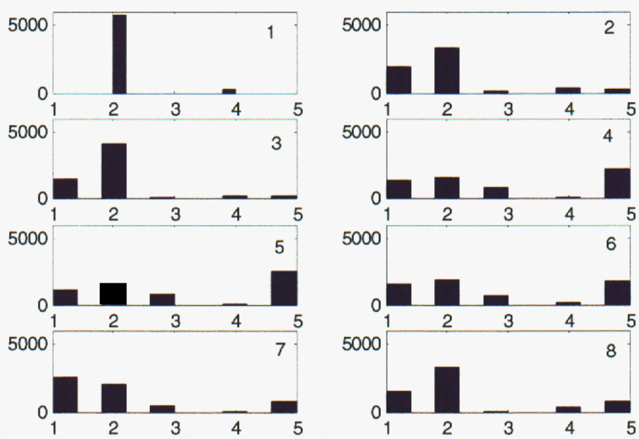

Figure 10. State Frequency (by Robot): Classification Tree From K-Means Clustering.

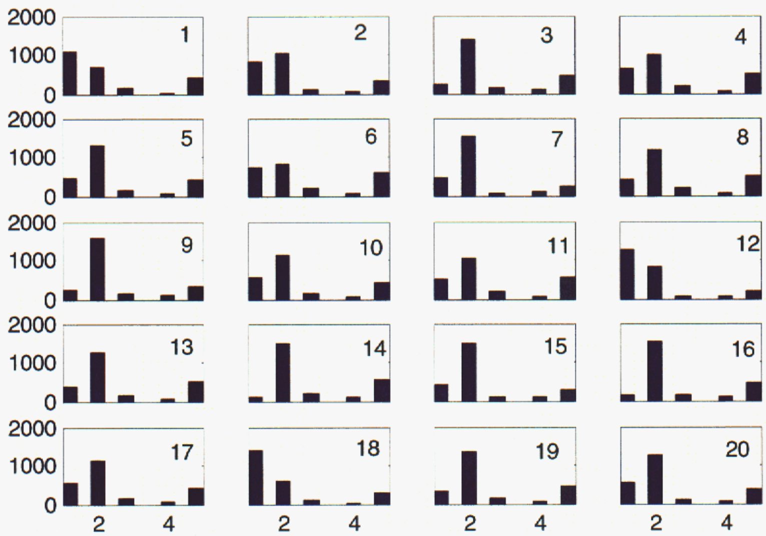

Figure 11. State Frequency (by Run): Classification Tree From K-Means Clustering.

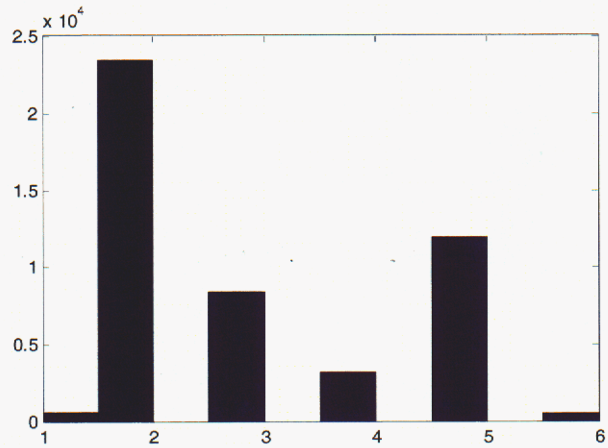

Figure 12. State Frequency: Using Classification Tree Developed From DIANA Clustering.

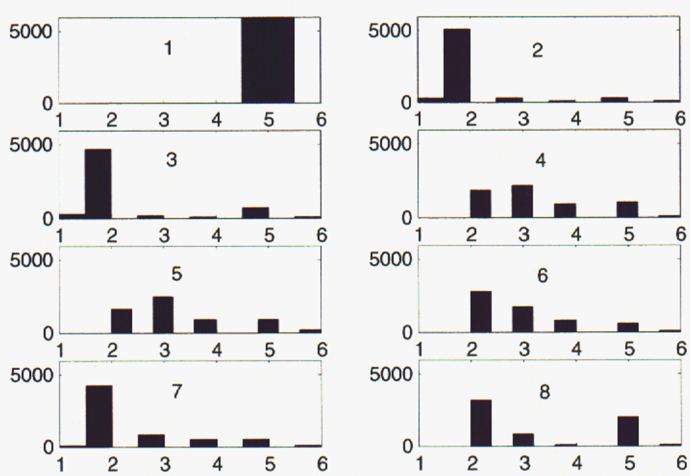

Figure 13. State Frequency (by Robot): Classification Tree From DIANA Clustering.

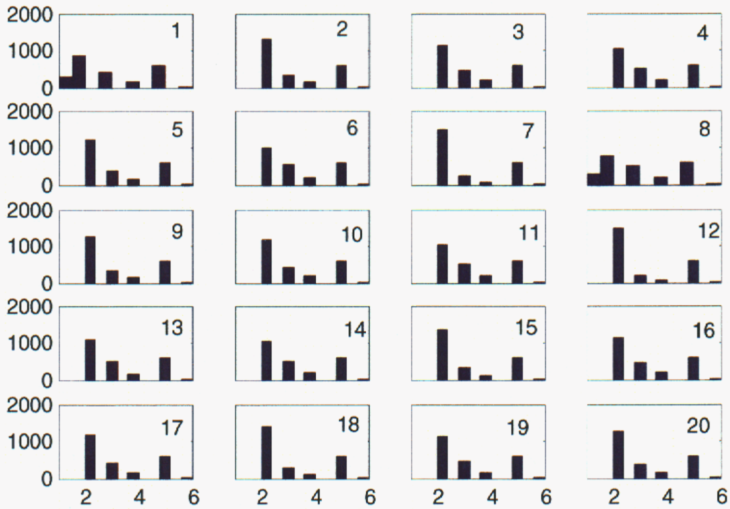

Figure 14. State Frequency (by Run): Classification Tree From DIANA Clustering.

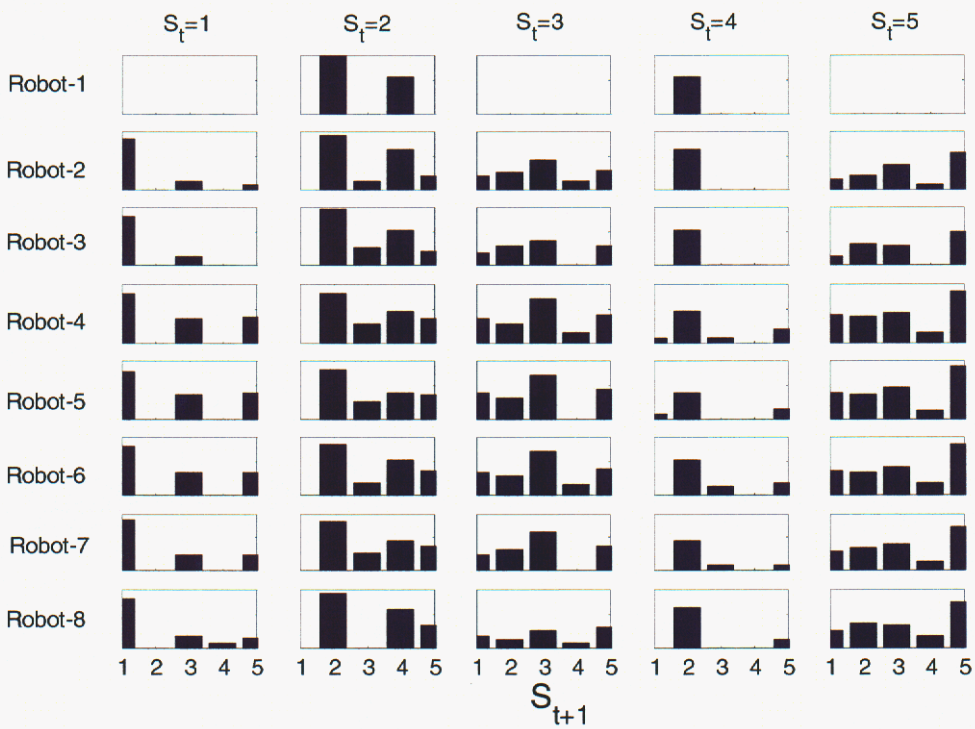

Figure 15. Log (Frequency+1) of Transitions from S_t to S_{t+1} for K-Means Analysis.

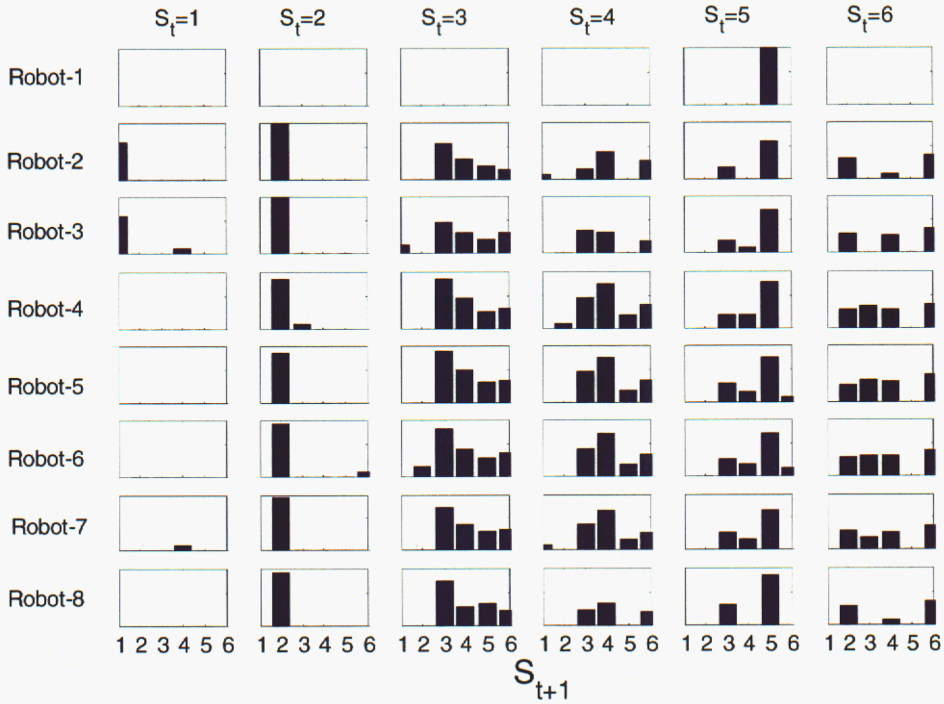

Figure 16. Log (Frequency+1) of Transitions from S_t to S_{t+1} for DIANA Analysis.

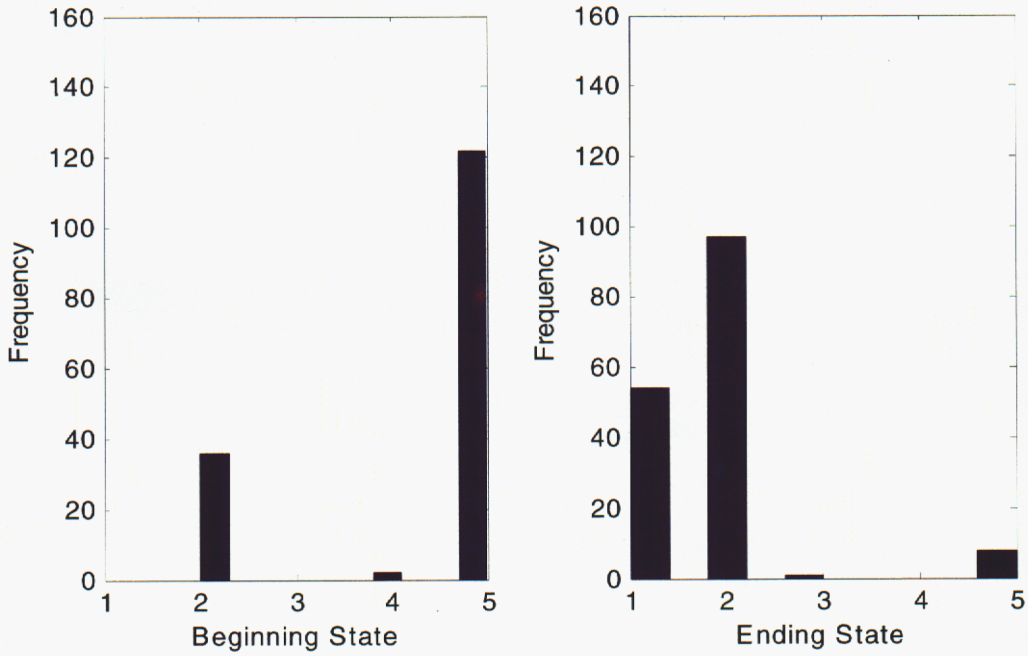

Figure 17. Beginning and Ending Distributions of States: K-Means State Space.

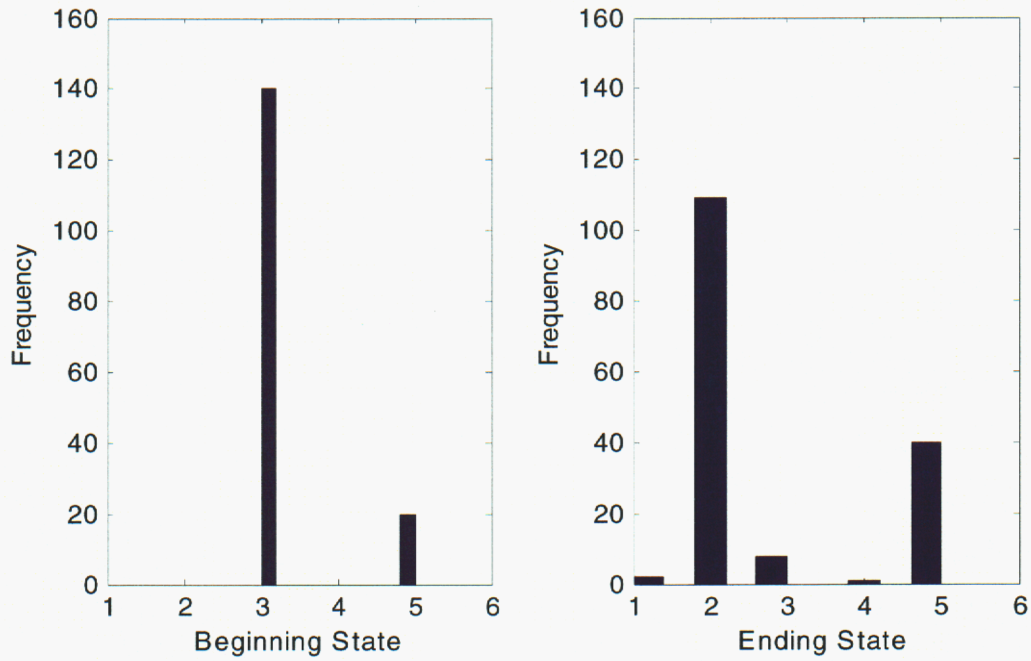

Figure 18. Beginning and Ending Distributions of States: DIANA State Space.

3.2 Project Year 2: Social Interaction

In the second project year, the objective was to develop and demonstrate mechanisms for storage and retrieval of episodes from episodic memory. For this demonstration an application was chosen that allowed ideas to be explored concerning the use of "primitives" as a basis for a universal representation of knowledge. In particular, emphasize was placed on modeling the social interactions of a group of individuals.

In identifying these primitives, emphasis was placed on research providing an ethogram of social behaviors developed through observations of chimpanzee colonies. This ethogram has served as a primary source of primitives (de Waal, 1989). This source has been chosen due to the complexity of chimpanzee social behavior and the availability of detailed behavioral accounts that, due to the observational methodologies employed, provide a level of objectivity that is not available for humans.

Two types of primitives have been defined. First, there are concepts that may be represented within a semantic network. Second, there are situations. These situations are composed of concepts that have been identified as primitives. Situations consist of actions placed in the relatively rich context created by associative relationships. The actions embodied by the situations each have goals associated with them. Goals generally reflect changes in drive states, as well as activation of other concepts. Appendix B provides a summary of the initial situation and concept primitives that have been identified.

For now, it is assumed that mechanisms exist for the recognition of perceptual cues that lead to activation of semantic concepts. As described here, each entity knows the other entities and, consequently, there is a semantic representation of each entity. Associated with the semantic representations of entities, there are attributes of each entity. Likewise, there are semantic representations for the behavioral states of entities. Thus, for the entity that is the subject of the threat, the semantic activation shown in Figure 19 may be anticipated.

As illustrated by Figure 19, Entity 1 is known to be dominant or more powerful. Dominance is an important concept in that it strongly influences behavioral responses to situations. The concept of dominance is treated as some combination of inherent strength and status, with status based on the outcome of recent confrontations. Entity 1 performs behavior that is perceived as either a threat or an attack. In actuality, based on the actual behavior and identity of Entity 1, there should be differential activation of these concepts.

Next, there is a second entity, Entity 2; and awareness of this entity produces activation of the concept "Guardian." Recognition of the situation "Seek Refuge" prompts the behavior of requesting that the guardian provide refuge. Based on the intensity of the threat/attack, the request for refuge, and the identity of the individual making the request, Entity 2 (the guardian) should experience differential activation of each concept with there being some likelihood for recognition of the situation as "Seek Refuge." Assuming recognition of this situation occurs and the guardian provides refuge, based on the intensity of this behavior and the identity of the guardian (Entity 1), the source of the threat or attack should experience differential activation of the concept "Provide Refuge." Based on this activation, the situation "Seek Refuge" may be recognized by the assailant and the threat or attack suspended.

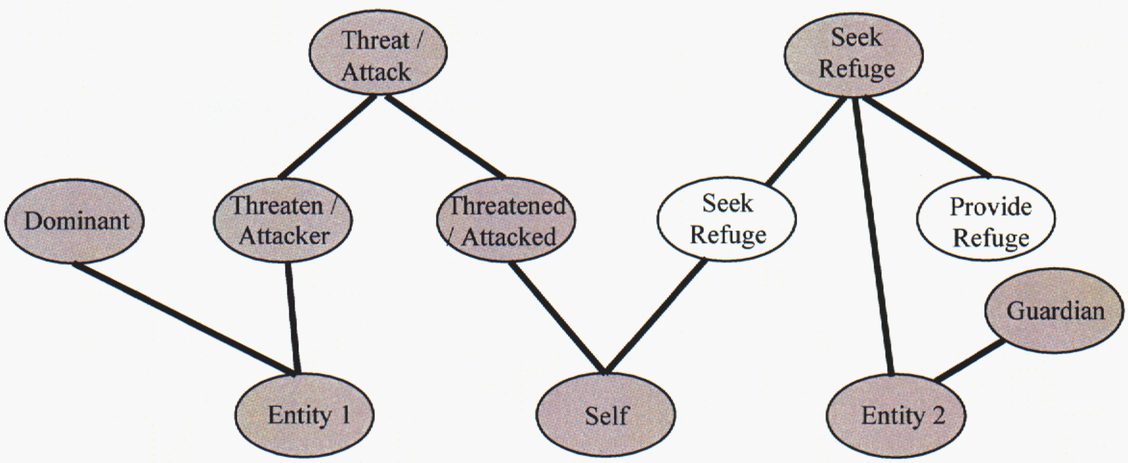

Figure 19. Semantic Activation for Entity Receiving Threat in Seek Refuge Situation.

For the demonstration, the subset of cues and situations listed in Table 2 was selected. Using a population of five entities with differential levels of dominance, a collection of interactions was simulated. The product of the simulation consisted of a level of activation for one or more cues for each of 1,900 individual instances. These levels of activation provided input to the cognitive model, populating its episodic memory.

Table 2. Cues and Situations Utilized for the Social Interactions Demonstration.

Concepts	Situations
entities	display
dominant	submit
weaker	usurp
threat	resist
attack	succumb
submit	displace
withdraw	recognize weakness
contender	keeping up appearances
defeat	concealment
resource	retribution
injured-fatigued	seek refuge
keep up appearances	provide refuge
avoid	suspend
seek refuge	
guardian	
provide refuge	
charge	

In the demonstration, a sequence of actor interactions involving the concepts and situations of Table 2 were presented to the cognitive model. A detailed time history of the dynamic activation levels of the observed concepts and situations was recorded into a (large) file. Once the complete sequence of actor events had been presented to the model, the recorded time history log file was ready for post-processing to construct a usable episodic memory.

The post-processing step of the time history log results in a very ordered sequence of episodes. This is accomplished by monitoring the "life line" of each unique concept instance or recognized situation. The "birth" time, "death" time, initial activation and maximum activation levels are extracted for each of these unique traces. The traces are sorted by their birth and death time such that traces that share an overlapping time period are located together in the episodic memory file. This is the definition of an episode in this computational instantiation of episodic memory: all traces that overlap in time.

Given any trace, all other traces that overlap any part of the time span of the given trace are found using the following construction. The temporal intersection of all traces that overlap the selected trace time span is expanded to find all traces that intersect that (larger, inclusive) time span. This time expansion is repeated until no more traces are being added to the inclusive set. This found inclusive set, given the initial trace, is the episode that contains that trace. Consequently, episodes in episodic memory do not overlap in time. This is how episodic memory is formed from the time history log of concept instances and recognized situations.

The generated episodic memory was searched and interrogated such that the episode containing a specified concept instance or recognized situation with the largest activation level was located in episodic memory. This was done using a very cryptic request like the following:

searchStr { s PRO-REFUGE (2} }

which meant: search for a situation where actor 2 provided refuge to some other (unspecified) actor. To perform the search, the structure and even content of the actual episodic memory file records had to be known by the requestor. Once the episode containing the trace with the most active situation of actor 2 providing refuge to actor X is found, that entire episode was presented to the user. After examining the episode content, the next episode or the previous episode in episodic memory was recalled and presented to the user. One way to think about this type of episodic review is to consider it as an example of interrogation. "Remember a time when X happened. What happened just before then? What happened after?"

Recalled episodes were fed back into the current cognitive process. In the demonstration, this resulted in a certain set of display components (one colorful grid cell per concept or situation) being colored according to their activation level as recalled from episodic memory (i.e., episodic recall).

The following is an excerpt from the time history file recorded during one demonstration session. Note that the time steps between recording events are not constant. Pay particular attention to the SUCCUMB concept activation levels, which grow from a value of 2.00897 to 2.70614 during this time span of 1431.63 to 1431.75 seconds. The second highlighted SUCCUMB is actually a recognized situation, and its raw value of 2.88043 is actually driving the raw value of the following SUCCUMB concept instance.

```
1431.63 Concept Instance Updates:
{SUCCUMB  (2 3) 2.01219 2.00897}
1431.66 Situtation Instances:
{SUBMIT (3 2) 4.21282 {inj-fatig  (3) 0.953491 1.11342} 3.34026 {DISPLAY-
DOMINANCE  (2 3) 3.22752 3.21415} 12.8566}
{SEEK-REFUGE (3 -1 2) 3.75407 {inj-fatig  (3) 0.953491 1.11342} 2.22684
{DISPLAY-DOMINANCE  (2 3) 3.22752 3.21415} 9.64246 {DISPLACE  (2 3) 3.61794
3.55618} 10.6685 {RETRI-1  (2 3) 4.24191 3.87764} 11.6329}
{RESIST (2 3) 2.88043 {USURP  (3 2) 3.00045 2.88043} 11.5217}
{SUCCUMB (2 3) 2.88043 {USURP  (3 2) 3.00045 2.88043} 11.5217}
{RETRI-1 (2 3) 2.2402 {CONCEAL  (3 2) 2.27293 2.26929} 9.07717}
1431.66 Concept Instance Updates:
{RECOG-WKNSS  (3 2) 2.84485 2.27588}
1431.67 Concept Instance Updates:
{RETRI-1  (2 3) 2.2402 3.05892}
1431.67 Concept Instance Updates:
{DISPLACE  (2 3) 3.61794 3.56074}
1431.68 Concept Instance Updates:
{USURP  (3 2) 3.00045 2.97645}
1431.68 Concept Instance Updates:
{DISPLAY-DOMINANCE  (2 3) 3.22752 3.21828}
 {SEEK-REFUGE  (3 -1 2) 3.75407 2.93815}
 {SEEK-REFUGE  (3 2 -1) 2.87582 2.85281}
1431.73 Concept Instance Updates:
{CONCEAL  (3 2) 2.27293 2.2722}
```

```
1431.73 Concept Instance Updates:
{inj-fatig  (3) 0.953491 1.11349}
1431.75 Concept Instance Updates:
{SUCCUMB  (2 3) 2.88043 2.70614}
```

The purpose of the post-processing of the time history file is to unravel each unique trace recorded during the "learning" or "watching" phase of the demo. Once the episodic memory had been constructed, episodes were located using the temporal intersection method described above. An excerpt from the processed episodic memory file is given below.

```
487.566 487.767 s SUBMIT (3 2) 4.63361  {dominat (2) 0 0.16273} 0.488191
   {DISPLAY-DOMINANCE (2 3) 2.10084 2.35738} 9.42951

491.185 500.354 ci inj-fatig (3) 0.953491 2.34778
491.185 500.724 ci weaker (3 2) 0.895752 2.93719
491.185 501.288 ci charge (3 2) 0.6651 3.00851
491.185 501.403 ci dominat (2) 0.406799 2.96959
491.185 501.754 ci pro-refuge (2 3) 0.900024 3.7016
491.185 502.293 ci guardian (2 3) 0.794922 4.80287
491.386 500.16 ci contender (3) 0.346954 1.42116
491.586 500.028 s CONCEAL (3 2) 7.08814  {dominat (2) 0.406799 2.96959}
   3.93219 {weaker (3 2) 0.895752 2.93719} 4.77019
491.586 500.229 s DISPLAY-DOMINANCE (2 3) 10.345  {dominat (2) 0.406799
   2.96959} 5.24292 {weaker (3 2) 0.895752 2.93719} 4.77019
491.586 500.631 s PRO-REFUGE (2 3 -1) 12.32  {dominat (2) 0.406799 2.96959}
   2.62146 {weaker (3 2) 0.895752 2.93719} 3.18013 {inj-fatig (3) 0.953491
   2.34778} 1.56734 {guardian (2 3) 0.794922 4.80287} 8.69767 {pro-refuge (2
   3) 0.900024 3.7016} 7.47693
491.586 500.832 s RESIST (2 3) 20.1144  {dominat (2) 0.406799 2.96959}
   5.24292 {contender (3) 0.346954 2.48299} 1.86698
491.586 500.832 s SUCCUMB (2 3) 20.1144  {dominat (2) 0.406799 2.96959}
   5.24292 {contender (3) 0.346954 2.48299} 1.86698
491.587 500.632 ci CONCEAL (3 2) 2.27293 5.57943
491.587 500.833 ci DISPLACE (2 3) 3.61794 6.67823
491.587 500.833 ci DISPLAY-DOMINANCE (2 3) 3.22752 6.71367
491.587 501.436 ci SUCCUMB (2 3) 2.01219 8.70847
491.787 499.827 s KEEP-UP-APPR (2 3) 6.43082  {dominat (2) 0.406799 2.96959}
   3.28171 {weaker (3 2) 0.895752 2.93719} 4.079 {contender (3) 0.346954
   1.42116} 2.87068
491.787 500.028 s SEEK-REFUGE (3 2 -1) 5.64249  {dominat (2) 0.406799
   2.96959} 3.28171 {weaker (3 2) 0.895752 2.93719} 6.1185 {inj-fatig (3)
   0.953491 2.34778} 3.71366 {guardian (2 3) 0.794922 4.80287} 9.97042
491.787 500.028 s USURP (3 2) 7.29671  {dominat (2) 0.406799 2.96959} 4.92256
   {contender (3) 0.346954 1.42116} 3.82758
491.787 500.631 s RETRI-1 (2 3) 13.346  {dominat (2) 0.406799 2.96959}
   4.92256 {weaker (3 2) 0.895752 2.93719} 6.1185 {CONCEAL (3 2) 2.27293
   5.57943} 8.72805
491.788 500.632 ci SEEK-REFUGE (3 2 -1) 2.87582 4.89514
491.788 500.833 ci USURP (3 2) 3.00045 5.66643
491.788 501.235 ci RETRI-1 (2 3) 4.24191 7.47781
491.989 500.23 ci RECOG-WKNSS (3 2) 2.84485 4.27805
500.832 500.832 s SUBMIT (3 2) 2.49065  {dominat (2) 0 0.187362} 0.562085
   {DISPLAY-DOMINANCE (2 3) 0 0.732854} 2.93142

507.064 515.779 ci weaker (4 5) 0.542603 1.26056
507.064 516.106 ci dominat (5) 0.763275 1.74868
```

A blank line has been inserted just before and after the episode. This demonstrates the temporal disconnect between this episode and the two surrounding episodes in episodic memory. In this form, the first two numbers on each line represent the birth and death times of the trace, respectively. Other numbers on each line represent the initial and maximum activation level for each trace (as well as some other data not relevant here). In fact, this excerpt from the episodic memory file corresponds to the same series of events given in the time history example above. The careful reader will notice that the times do not match between these two samples. This is because the two files were captured during two different runs of the same input data set. They are equivalent regardless of the time bias between the two runs.

3.3 Project Year 3: Desktop Applications

In Year 3, we have created a computational model equipped with an episodic memory of a user interacting with Microsoft Outlook. In this way, the cognitive model is able to:

- interpret in real time which projects the user is working on
- detect when the user is engaged with new projects and/or people by comparing ongoing events to its situational knowledge and its episodic memory
- add situational knowledge and associative knowledge to its own architecture after querying the user regarding new projects and/or people.
- learn about these bits of information from a *tabula rasa* position—i.e., with no prior specific knowledge of people, projects, or email attachments.

Before describing the FY03 functionality, a brief review of the literature relevant to the learning mechanisms replicated in this application is in order. After this review, the functionality and technical details are presented.

While there are computational systems that are said to be able to learn (e.g., genetic programming), because of our human-centric orientation, we must ask the question "by what mechanisms do *humans* come to 'know' something?" Our answer to that question is basically that humans learn about things and thereby come to "know" them through experience. Learning is an empirical process and that process involves, at its core, dynamic context discrimination and categorization. Categorizing things is something humans do quite readily—so much so that we are quick to find recognizable patterns in things that are random—like seeing shapes and faces in cloud formations. Dynamic context discrimination and categorization are the primary ways by which we make sense of our world. We relate current experiences to things we have encountered in the past (e.g., Cantrambone and Holyoak, 1989; Gick and Holyoak, 1983; Hofstadter, 2001).

The psychological literature on analogical problem solving has a lot to say about this particular issue. Briefly, this literature describes the process by which humans use previously encountered information to solve a current problem. For example, a student of physics may use what she learned in algebra to set up and solve certain kinds of problems. That is, she recognizes that concepts learned in algebra are relevant to current physics problems she is trying to solve.

To this end, the analogy community has identified several stages, or subprocesses, involved in bringing previously encountered information to bear on current situations or problems. First, there is the process of *recognizing* the relevance of the past information to the current situation. This process of recognition also implies that that information is retrieved from memory. The second phase of analogy making is *mapping*, in which correspondences between aspects of the current problem and past experience are drawn. The third phase is *adaptation*, in which what is known of the previous situation is applied to the current situation. The fourth phase is *learning*. This is the phase in which a generalized version of the situation or problem class is formed such that a *schema* is created. This schema, therefore, is akin to an averaged version of a situation or problem class such that the key components of that class are preserved while situation-specific details drop away (Butterfield and Nelson, 1989; Cantrambone and Holyoak, 1989; Gentner and Markman, 1997; Gick and Holyoak, 1983; Hofstadter, 2001). An example of this is the schema you have for a fast-food restaurant. There are key components that all fast-food restaurants have in common regardless of brand name, type of food served, location of the restaurant, etc., that tell you that a particular establishment is an example of the fast-food genre. The cognitive framework does not explicitly address all four stages of analogy listed above. However, it does address two key aspects of analogy explicitly: retrieval of relevant information from memory and learning.

The genericized memories we all have, such as eating at a fast-food restaurant, are currently instantiated in the situation library portion of the model. These memories, or situations, are events that are common enough that they take on a script-like characteristic. While the memories that are written to the episodic module in the architecture do include instances of these generic situations when the model encounters them, it also includes a record of novel events—events that the model is not otherwise equipped to recognize as known situations. When the model encounters the same novel set of events a second time, it is able to retrieve the first instance of that set of events, recognize the relevance of the first to the second instance, and query the user about the exact nature of this new event type. The user response then guides the model regarding the creation of a new schema that is then represented in the model's situational knowledge library.

In this manner, a system with an integrated episodic memory augments human cognition in several ways. First, because the system learns and grows alongside the user, it can continue to perform augmentation tasks like discrepancy detection and acting like a decision aide long after the initial version of the model is created. Second, because recall from its episodic memory is flawless, it can augment the user's recall by allowing the user to query it about past events.

3.3.1 Functionality and Capabilities Developed During FY03

The initial state of the episodic memory-enabled cognitive model for this FY begins with no specific knowledge of the user's desktop projects or of people affiliated with those projects. The only concepts in the associative network of the model are five types of novel stimulus concept (one each for email headers To, CC, From, Subject, and Attachment) and action concepts (concepts that represent the commands used when interacting with the software and the actions surrounding the use of Outlook; e.g., email sent). Therefore, the model starts out as a blank slate with regard to the particular user's projects and colleagues.

As information is gleaned from Microsoft Outlook, the model is capable of recognizing instances of known concepts appearing in the To/CC/From fields, in the names of attached files, and in the subject line of emails. When the model encounters text in one of these fields that it does not recognize (i.e., that does not appear in its associative network), the corresponding novel stimulus concept fires in the associative network (e.g., if the unknown text string appears in the CC field, the "novel stimulus CC" concept fires), thereby allowing the model to recognize that it is unfamiliar with a particular string of text.

By creating multiple novel stimulus nodes, we essentially are enabling the model to learn finer-grained distinctions than it would be able to otherwise. For example, I might think differently of an email on which the boss is copied rather than one for which she appears in the From field. Likewise, I might classify the importance of an email differently if it has a subject "Final LDRD report" than I would an email with an attachment entitled "Final LDRD report." Furthermore, by enabling the model to make finer-grained distinctions, we enable it to create more specific generic categories of events—for example, all emails that our boss is copied on that have relevance to the episodic memory project are treated differently than the emails that our boss is copied on that are recommending us for an award.

Currently, when one of these five novel stimulus nodes fires, it allows the model to recognize a situation called "introspect," which then enables the model to create a new concept node in its associative network that represents the novel string of text that is the cause of the novel stimulus activation. If that string of text appears in the To/CC/From fields, the new node is created automatically. If the novel string of text appears in the attachment or subject information, the user is queried with regard to the importance of that string or a subset of that string in order to prevent an "all or nothing" approach to adding new subject information to the associative network. Regardless in which of the five fields the novel text string initially appears, once a concept node is created to represent that text string, it can be recognized as the same text string in any of the five fields.

When new information is detected in the To/CC/From fields, the user is queried about whether or not to pay attention to that text in that particular context. (This query does not have any impact on the creation of a new node in the associative network.) Right now, the model has no difficulty adding new people to projects; however, if the user indicates that a novel string of text does not belong to a given project, regardless of the field in which it appears, the model must be able to inhibit similar question about that string of text in that particular context in the future. This inhibitory capability has yet to be developed. As of now, the user is queried every time an unaffiliated string of text is encountered, even if it is in the same context as the initial encounter.

However, the "novel stimulus" concept will not be activated again for items in the To/CC/From field because the model will be able to recognize that string of text via the automatically generated concept node.

At this point, the model is not capable of automatically adding a known person to a known project—that is, if Carl Lippitt begins to work on a project the user has been working on for a while, the model will not automatically add Carl to that project. And the user must manually add the person's name in order for that association to be made.

Currently, the model makes use of the dynamic contents of episodic memory when it recalculates the weights that affiliate concepts differentially to projects. This process takes into account both the frequency of association between a given concept and project (e.g., Carl Lippitt and episodic memory), how many projects a given concept is affiliated with (e.g, Chris Forsythe is affiliated with all projects whereas Carl Lippitt is only affiliated with two), and the response that the user gives to instances of new concepts' potential affiliation with existing projects. For example, Kathleen Diegert will only be affiliated with the episodic memory project if her name is included in one of the five monitored fields and if the user indicates that she should be affiliated with episodic memory—if the user says "ignore this," Kathleen Diegert can be recognized but she will not be affiliated with any project via weights in the pattern recognition algorithms. In this application, weights ranged from 0 to 3 in these pattern recognition algorithms, with 0 indicating that the concept is affiliated with a project but is totally nondiagnostic of that project. The higher the weight, the more diagnostic that concept is of that particular project (situation).

In addition to weights decreasing with regard to decreased diagnosticity, there is an additional process that decreases weights as a function of the number of times a given concept has been affiliated with a given project in episodic memory, mimicking the process of *habituation* to an extent. In this way, a new addition to a project has a higher weight to that project than one that has been affiliated with that project from the beginning. In addition, when a new project is created, the weights on the concepts associated with that project start out at a maximum weight (which is currently 3), then those weights begin to decay as more and more information is collected on this project and its associated concepts. Currently, this decay is not a continuous scale, but degrades on a discrete scale—3 for brand new (i.e., less than two instances), 2 for two to five instances and 1 for more than five instances.

Additional capabilities developed in FY03 include:

1. In previous years, in order to query episodic memory about specific temporally related episodes, the raw temporal data that was generated by the initial processing of incoming information by the cognitive model had to be processed offline by specialized scripts designed to generate a log of temporally related events. Then, this resulting log could be used by the cognitive model to answer queries from the user regarding temporally related events. This year, we have developed the capability to generate a queriable log that is available in real-time. That is, the user can query the episodic memory about an event that just occurred as easily as about an event that occurred two weeks prior.
2. Previously, the search capabilities were somewhat limited in that in order to search episodic memory the user had to know specifics about the implementation of the episodic log. This year, all of the implementation specifics of the episodic log have been hidden and are accessed by specifying a query request that the episodic memory system uses to

find the best (or best N) matches to the request. The search can ask for any episodes that fall within a specified time range (i.e., what was happening on September 23, 2003, between 8:30 and 9:00 am). The search can find an episode based on the name, or partial name, of any concept used by the cognitive model (i.e., recall an episode where Chris Forsythe is mentioned). Episodes can be found based on the activation intensity level (i.e, recall the five most memorable episodes—those that have concepts with the largest activation intensity level). If an internal episodic memory index is known, it can be used to recover the entire episode that surrounds that event.

In the most general case, a query can be specified that will find the best N matches for a multi-part query where several simultaneous constraints are used to find the matching episodes. Any of the above-mentioned search methods can be combined into a single search request; i.e.,"What were all the messages sent to Chris Forsythe in September 2003 about the Insider Threat project that also included Ann Speed?"

3. The search has been extended with regard to the specificity of the questions that can be asked. Specifically, the user can specify
    a. things that are required (e.g., "I only want to know about emails I've sent to Carl Lippitt"),
    b. things that are desired (e.g., "I want to know about all emails regarding episodic memory, but I'm most interested in those that have Carl Lippitt in one of the To fields"),
    c. required time relationships (e.g., "I want to know about situations in which A happens then B then C, but I don't want to know about situations in which B happens then C then A"),
    d. desired time relationships (e.g., "I want to know about situations in which A, B, and C happen, but especially A then B then C"),
    e. and unscored query records (e.g., "I would like to have all emails relevant to episodic memory, but if Carl Lippitt is on those emails, I'm less interested in them").

4. In previous instantiations, the model could not interpret novel information. Current advances allow the model to (1) recognize when it does not recognize something (i.e., it does not recognize a name in the To field of an email), (2) determine where that novel stimulus most likely belongs in terms of its understanding of the world based on the other stimuli present in the same episode (i.e., it sees other names in the To field it recognizes and words in the subject line it recognizes and indicates which projects to which these names and words are related), and (3) to query the user regarding the accuracy of this relationship (i.e., it requests that the user indicate if that new name be added to one of the projects or if it should ignore the new name).

3.3.2 Building the Cognitive Model, Episodic Memory, and Learning Algorithms

The actual cognitive model as implemented in Umbra does not accept unknown concepts and situations. It assumes that the expert model, as defined by its several configuration files, are static and unchanging while it is running. However, the central thrust of this demonstration was to show learning by the cognitive model. Learning implies adding previously unknown concepts and situations to the model as it is running. To address this tension between what the cognitive model was designed to do and how it was to be used for this learning demonstration required an iterative learning approach. The model starts with one set of known concepts and situations and as new ones are learned the cognitive model is restarted to use an updated set of configuration files.

Outside of the cognitive model, per se, is an understanding of what a project "is." For this demonstration, a project is a defined association between people who send email to each other and mention certain keywords in these email interactions. The system is able to bootstrap itself from an initial state where, although it knows about projects, it does not know anything about any specific project. It does not know any people. It does not know any keywords. All it "knows" is the definition of a project and when it does not recognize a person or keyword. The system also knows that it can learn about people from the To, From, or CC fields of an email message. And, it knows that keywords can be found in the Subject field of an email message or in the name of an attached document to the email message.

Visual Basic for Applications (VBA) was used in Outlook to add the ability to generate perceptions to the cognitive model based on the content of sent and received email on the user's desktop PC. The VBA perception generator was told what to look for and where it was to look for it through a configuration file written by the OPAL controller code (implemented in Umbra through Tcl scripts). If this configuration file ever changed, the VBA perception generator would re-read the file and start using the new set of instructions. A typical example from this configuration file is given here.

```
cc|to|from/dude_Tegnelia,JamesA/Tegnelia, James A
sub|attach/kw_DC/DC|Daimler Chrysler
```

This example was taken from the final configuration file generated while running the demo. It states that if the phrase "Tegnelia, James A" is noticed in either the CC, To, or From field of any email message, that the perception called "dude_Tegnelia,JamesA" is to be sent to the cognitive model. Also, if the phrase "Daimler Chrysler" or "DC" is noticed in either the Subject field or in the name of an attachment, then the "kw_DC" perception is to be sent to the cognitive model.

The perception generator knows nothing about projects. It only knows how to find text phrases (case insensitive) in specified locations of email messages. If the perception generator scans a field of an email message and does not find any text string that it recognizes, then it emits a "novel" perception. A novel perception for each type of email source data is generated if none of the specified searches in the configuration file match the content of that email source field. The perception generator is, in effect, telling the cognitive model, "I see something, but I don't know how to see it." In this case, the perception generator also passes along the text of the field that it observed but could not scan to the cognitive model.

To send these perceptions to the cognitive model, the VBA perception generator writes its output into a text file at a known location in the file system of the host PC. Tcl procedures in the OPAL Umbra code are configured to tail (watch the new content being added) this text file. This reader code converts the perceptions into concept instances and feeds each concept instance into the cognitive model. After three seconds have elapsed, this reader code then removes the concept instance from the cognitive model by setting its raw activation level to zero.

The reader, when it receives a novel perception, generates a novel stimulus. It also attaches the unrecognized content of the text string given to it by the perception generator to the novel stimulus concept instance. For this demo, the cognitive model was modified to allow the attachment of arbitrary textual content to any concept. This was required so that future episodic recall of the novel concept would also provide the raw text of the email field that caused the novel concept to occur. Any question about this novel stimulus would have to be mined from episodic memory if it was to be used by the model at any time in the future.

Unlike the episodic memory demonstration for FY02, the episodic content of episodic memory is generated in real time while the cognitive model is operational. Traces of concept instances and recognized situations that overlap in time are defined to be an episode. Once all active traces have died out, the episode is closed and written to episodic memory. While some trace remains active, the current episode is said to be in an unfinished state. An unfinished episode cannot be found by an episodic query. However, episodic memory does have a "reflect" function that always returns the last episode in episodic memory, even if that episode is "unfinished."

When a novel perception is presented to the cognitive model, the fact that the perception occurred is recorded in episodic memory as part of the episode of other active traces which other parts of the email message had activated. From the initial condition, there is only one situation that the cognitive model can recognize. This is how that situation is stated to the cognitive model:

```
s introspect 1
sc introspect 5 novel_from 1.0 0 novel_to 1.0 0 novel_cc 1.0 0 novel_sub 1.1
   0 novel_attach 1.0 0
```

Which means: there is a situation, called "introspect." The introspect situation is generated by summing the weighted activation level of five concepts: novel_from, novel_to, novel_cc, novel_sub, and novel_attach. It is constructed in such a manner that whenever at least one of these novel concepts are active then the introspect situation is recognized.

It so happens then, when a situation is recognized that also has the same "name" as a known concept, that the concept of the same name is activated with the activation level of the situation. This is a kind of feedback at the cognitive level, from situations to concepts, and allows the generation of situations based on a hierarchy of other recognized situations. The demo does not use this hierarchy, but the fact that a recognized situation caused a concept to be created is used. When this occurs, the cognitive model also provides a mechanism to allow Tcl procedures, not directly associated with the real-time operation of the cognitive model, to operate at the time of this event. What the demo does at this event is very simple. Unless it is inhibited from doing so (discussed below), it remembers a reference to the current unfinished episode and prints a simple question to the user in a text window: "Excuse me, please. May I ask you a question?" Something novel, unique has occurred and the model would like to resolve the uniqueness, if possible so that, if this happens again in the future, it could recognize it and assign it some meaning, some association to the projects being tracked.

Several of these "requests" can be queued, waiting for action by the user. When the user is ready to respond to the request generated by some novel stimulus, he indicates this to the code by using the procedure called "ask."

Ask, as a Tcl procedure, removes the oldest request from the question queue. An "ask-user" concept is fed into the running cognitive model to log into episodic memory the fact that the assistant is now asking a question about a specific novel stimulus. The raw data of this "ask_user" concept contains a reference to the episode in question. Then the episodic memory is queried to recall the episode and extract from the episode all of the known data (references to known concepts—people and keyword phrases—and recognized project situations) and the raw data text string that was attached to each piece of novel concepts. There can be several known and several unknown things about any episode. The known and unknown data are presented to the user to both spark his own recollection of the email in question and to illicit some response to the novel data from the user.

The user can choose to ignore the novel data. To do this they say either "ok" or "*nada*" in response to the question. Or the user can use several Tcl procedures to inspect the current state of the project knowledge database. If the user makes any additions or modifications to the project knowledge database, other concept instances that reflect these user actions are presented to the running cognitive model to log these events into episodic memory as well. The design of this portion of the demo is to allow the user an opportunity to state that the email really belongs to a new project, or that some new keyword or person needs to be added to an existing project so that future email that has similar content or recipients will cause the correct project to be recognized by the cognitive model. In any case, once the user is satisfied with the present state of the project knowledge database, he is to indicate this by using the Tcl procedure named "ok." The user's response also generates a concept instance that is fed into the running cognitive model to log these events into episodic memory.

Adding the assistant questions and user responses and modification events into the content of episodic memory allows for future introspection of the episodes to "remember" if the user did anything in response to a certain novel cue. Indications that the user did not care (he answered "ok" or "*nada*" without making any project modifications) about the novel cue was intended to cause the construction of an inhibiting situation recognizer for future novel concepts. The purpose of the inhibitor is to stop the automatic asking of a question ("Excuse me, please . . .") if the current episode is very similar to some previous episode (or sequence of episodes) where the user had chosen to ignore the question. As variations of the novel data and components are also ignored by the user when asked, the several answers and their variance in content will allow a relaxation of the initial specific constraints so that, over time, the inhibitor becomes more and more generic, and thus inhibits more and more questions, having learned that the user does not care about similar messages.

As each novel episode is presented to the user, any novel concept that would refer to a person (novel_to, novel_from, novel_cc) is automatically converted into a concept for that person's name. If there is a recognized project situation and some of the people referenced in the email message are not associated with that project in the internal project knowledge database, then the user is also asked for permission to add these users to that project activating cues to recognize that project in the future.

Once all questions have been answered by the user and the question queue becomes empty, then the quasi-static configuration files of the cognitive model are updated, and the cognitive model is stopped and restarted to use the new, updated configuration files. Perceptions that had previously caused some novel concept to be generated will now cause a learned concept to be recognized (if either automatic or manual modifications to the internal project knowledge database had occurred). The weights assigned to the concepts that activate a given project situation recognizer are computed as a function of their relevance to the project itself. This is done by scanning the internal project knowledge database to determine how diagnostic the person or keyword is for that project (are they referenced by more than one project). In addition, each keyword or person is used to scan all of episodic memory for an indication of how diagnostic that keyword or person has been in the past for that project. As time goes by and more and more episodes are recorded in episodic memory and more and more knowledge is gleaned from these episodes and the user's responses to novel cues, the weights tend to capture the long-term "expert" knowledge that associates people and keywords to that particular project.

3.3.3 Model Validation

The first method of model validation for FY03 is to demonstrate that after a given amount of time and input the model is sufficiently different than it was at its initial instantiation. One obvious way to do this is to look at the things that the model learned over the time course of initial operation. The following measures represent change in the model over a period of 25 hours (three working days). There were a total of 10 emails sent and 24 received.

First, in terms of the number of nodes that were added, the model started with only a generic notion of emails and of interactions with emails. Specifically, the concepts the model understood were:

1. Novel text in From field
2. Novel text in To field
3. Novel text in CC field
4. Novel text in Subject field
5. Novel text in name of attachment
6. Email was sent
7. Email was received
8. Ask user a question
9. User tells episodic memory to ignore novel text
10. User tells episodic memory about something new (person, project, keyword)
11. User tells episodic memory to add something (person, project, keyword)
12. User tells episodic memory to quit asking questions

The concepts the model added during the course of 25 hours were the names of 33 senders or recipients of email and 10 key words. All of these concepts appear in the associative network.

In terms of adding situations to the context library, the model went from having a single situation, "introspect," to having six situations, five of which represented different projects. The introspect situation is recognized when the model sees a string of text it does not recognize in one of the five email fields. When the model recognizes the introspect situation, it queries the user about that string of text within the context of the projects it knows about. This was aided by the model querying the user a total of 22 times during the 25-hour period. At this point, we have not collected enough data to demonstrate that the ratio of questions-to-emails-sent-and received changes as a function of time.

The five new situations the model added are below.

```
s_proj_CognitiveModeling
sc proj_CognitiveModeling dude_George,Vivian dude_Abbott,RobertG
dude_Forsythe,JChris kw_CognitiveModeling kw_HumanFactorsEngineering

s_proj_EpisodicMemory
sc proj_EpisodicMemory dude_Speed,AnnE kw_Umbra kw_epimem kw_EpisodicMemory

s_proj_Daimler
sc proj_Daimler kw_DC
```

```
s_proj_JSBsim
sc proj_JSBsim dude_jsbsim-devel@lists.sourceforge.net kw_jsbsim-devel

s_proj_Misc
sc proj_Misc dude_NWCOnline dude_StarwoodPreferredGuest dude_healthupdate
dude_.NETInsight kw_NetworkComputing kw_SandiaDailyNews
```

The fact that the model learned a total of 43 new concepts and only 5 new projects is of note. As will be discussed in the next section, a bit of software developed by another company to do what episodic memory does has difficulty with prioritizing and with filtering information so that it only presents relevant data to the user. By interacting with the user, the episodic memory finds out what is and is not important to the user, and it is able to tailor not only the information it presents to the user, but the information it pays attention to itself. This is evident in the fact that only a small amount of the total information contacted by the episodic memory was considered by the user as being important enough to include in the model itself.

3.4 Discussion

The notion of having a technology that can augment a human's use of a computer system is by no means a new one, and the episodic memory software developed during this project is by no means the only one that has been developed recently. Below, two such technologies are reviewed and compared with the current technology.

3.4.1 Brief Review of Similar Technologies

3.4.1.1 LifeStream and TOWER

The current project applied technology utilizing computational cognitive modeling including episodic memory to the problem of managing documents and email on an individual's desktop computer. The application tracks documents created by the user and infers the projects to which the different documents are linked and which people are working on which projects. The application also responds to queries posed by the user such as, "What was the most recent email I sent to John Doe regarding Project X?" or "What projects did I start in the month of November of 2002?"

While there are technologies that exist that create detailed histories of a user's interactions with his computer (e.g., LifeStream) and with team members functioning from various locations (e.g., TOWER), these technologies are fundamentally different from the current project in two related ways. First, the ultimate functionality of these alternative technologies, while very similar, are not based on a psychologically plausible model of the user's understanding of his computer and his interactions with his computer. So, while LifeStream includes helper agents that usher different bits of information around, those agents are not created to behave the way the user would.

Likewise, TOWER, which is intended to augment social interactions between geographically separate team members, focuses on a 3-D visual interface rather than on a human emulation engine to drive and direct interactions such that the activities of different users are easily monitored within a virtual space. While TOWER clearly has its advantages in terms of augmenting ambient awareness of other team members, its focus on augmenting human-human interaction is much different from that of the current project. The software mediates human-human interaction in that it makes the activities and presence of other team members more salient. It does not attempt to overcome the problems of user-machine misunderstanding and frustration, or even user-user misunderstanding. We believe that powering this functionality via a psychologically plausible human emulator is key not only for the effectiveness and usability of the technology but also for its eventual scalability and generalization beyond the desktop environment.

Second, these technologies are not structured in a manner that allows them to learn. They are able to track documents, projects, email, and resources, but they are not able to compile this information in a manner that allows them to flexibly adapt to the individual user .

3.4.1.2. Six Degrees by Creo

Of more interest is the comparison between the current project and software by Creo called Six Degrees which won the Innovator of the Year Award at the Comdex Exposition in 2002. The similarities between Six Degrees and Episodic Memory include an ability to quickly search through the corpus of emails in the user's Outlook folders to find all emails, attachments, and people that are associated with one another. In particular, Six Degrees offers the ability to not only perform this correlation on emails after the installation of the software—it is able to reconstruct an entire history of emails, attachments, and people by gaining access to the user's entire Outlook file structure, thereby enabling it to retroactively correlate people, attachments, and emails. The current version of the Episodic Memory software, on the other hand, is only capable of making these correlations going forward from the time of model initialization on the user's hard drive.

The method by which Six Degrees performs this people/attachment/email correlation is a significant point of departure for the two software packages. An example will best illustrate the Six Degrees method of correlation. I can search for all emails associated with a given person by entering her name in the Legend field of the Six Degrees interface. Six Degrees will then display (1) all other people affiliated with this person (i.e., all others who have sent or received emails on which they were a sender or recipient), (2) all attachments sent to them (or sent to others via an email that the person also received), and (3) all emails sent to or received from the person. A user may view these three types of information either in conjunction with one another in three tiled windows or as separate windows in the interface.

Likewise, the user can select a particular document that was sent as an attachment to the person for examination by the software and will see all related attachments (i.e., attachments with identical names), the people who received these attachments, and the emails that contained these attachments as lists of items in separate windows. Note that this information appears as three separate lists (even when the windows are tiled and, therefore, simultaneously visible) and that it is not made clear who received which emails and which attachments were in which emails.

If a given person has not been included in emails very often or if a given document has not been attached to many emails and sent to many people, this interface provides valuable, easily searchable information. However, if a person has been included in many emails—especially if those emails are relevant to several projects or if a given document has been widely circulated with regard to several projects or applications—the information provided in this interface becomes unwieldy fairly quickly. This was illustrated by a case in which the person had only been known approximately 30 days and a search to find other people with whom this person was associated yielded 31 names in the People field, 11 of which were not recognized.

Furthermore, the emails displayed that were related to the individual are listed according to "relevance," where relevance is not defined by the user but is instead defined by recency or "common-ness." Therefore, the most relevant emails listed in the interface were notes sent to this person about a social function rather than those relevant to the Episodic Memory project.

Rather than looking at names or documents per se, Six Degrees also allows the user to search by keyword. For example, by typing "episodic memory" into the Legend field, all emails that have that phrase somewhere in the subject line are displayed, along with all attachments with that phrase in their file names. However, there are no people displayed as being relevant to the phrase "episodic memory," which is the name of a project that is actually affiliated with five people. Furthermore, only the attachments with the exact phrase "episodic memory" in their file names are displayed even though there are other attachments that have been sent that are relevant to the episodic memory project.

In short, while Six Degrees performs the task of filtering emails, attachments, and people quickly and efficiently, it does not provide a filtering ability that is specific enough to display people, attachments, and emails by project (given that a project has multiple keywords, multiple documents, multiple people). In other words, it only searches on one factor—that is, the user cannot perform the search "episodic memory" and "Mark Schaller." Furthermore, the Six Degrees search seems to be fairly literal. If the user types in "episodic," the results are different from those for a search on "episodic memory." The software does not recognize the equivalence between "episodic" and "episodic memory," nor does it recognize that "epimem" is a nickname for "episodic memory," nor is it equipped to learn these equivalencies. Finally, Six Degrees keeps track of all of this information in a passive way. There is no ability for the software to make deductions about the relationships between people, attachments, and emails based on the user's behavior and to then query the user to validate its deductions. Rather, it provides a quick if literal and inflexible method for navigating a large amount of information in Outlook.

By way of comparison, the functionality developed through the current project provides a capability to address many of these issues. Correlations between documents, people, and emails are done via project[2]. A project is defined by way of key words that appear in attachment file

---

[2] Even though the current version of the software deduces these relationships according to project, this is not a requirement for the software. Defining these relationships by project is actually an emergent property of the functionality – the software itself "knows" nothing about projects per se; therefore, the organizing framework could be any number of variables such as names, organizations, fiscal years, companies, etc. In short, the organizing factor can and would be user-defined.

names and email subject lines. They are also defined by watching where the user files documents and emails in Outlook folders as well as on the desktop.
Furthermore, it performs these correlations in an active, human-like way such that as the software runs, its "mental model" of the user's activities on his desktop continues to resemble the user's mental model, which enables it to make more specific correlations between people, documents, and emails—all within the overarching framework of projects.

Not only can the software respond to queries like "Mark Schaller," it can respond to a query like "the most recent email sent to Mark Schaller regarding Episodic Memory." In addition, when the key phrase "episodic memory" is entered as a query, it can produce the names of people who have received emails relevant to the Episodic Memory project, it can list attachments that exist relevant to the project, and it can display relevant emails—even if the term "episodic memory" has not appeared explicitly in conjunction with these particular people, documents, or emails. When the software notices that a new set of words is being used in email subject lines, regardless of the people who are receiving the email, it can query the user regarding these words or phrases and regarding the people receiving/sending the email, with the goal of determining if a new project has been started and, if so, who is involved, key phrases etc. In the event that these new phrases are not indicative of a new project, the software is able to create equivalence classes, thereby enabling the model to recognize that words such as "epimem" and "episodic memory" refer to the same project and should be treated as equivalent keywords.

Similarly, when the software notices that a group of people affiliated with a given project are now receiving emails with an unrecognized subject line, it can query the user regarding the meaning of this subject line and if that word or phrase is relevant to a known project or if a new project has been started with this same group of people.
It is important to make explicit that the Episodic Memory is not a method for conducting n-way searches. Rather, this software is an active participant (and observer) in the user's experience. It maintains a dynamic representation of relationships between documents, people, emails, and projects that will eventually be accessible to the user for visual inspection—this representation is akin to a network diagram or concept map—and the user will be able to verify the software's understanding of the user's projects by actually looking at the software's representation in addition to asking the software things such as, "What people are affiliated with the Episodic Memory project?" In addition, because the theoretical rationale for the model is based on the psychological literature, the model has a human-like memory for past events—a personal history of the user's actions within the desktop environment that the user can query regarding specific events (e.g., emails sent to Mark Schaller regarding epimem functionality) and regarding the surrounding events (e.g., "What did I do right before I sent that email to Mark? What did I do immediately after?").

3.4.2 Future Possibilities for Additional Episodic Memory Functionality

The current instantiation of episodic memory, while an effective and important addition to the model, could include additional functionality. Listed below are some ideas about future research into additional episodic memory functionality.

- Include an ability for the model to create episodes out of events that are temporally distant from one another, as in email threads. Action at a temporal distance is an important capability for the model because it allows the model to have a more global, or high-level understanding of the user and it lays the foundation for determination of causality for events that are not temporally immediate to one another. Include comparator as an analogy-maker to enhance model's ability to leverage past experiences for current situations and to enhance learning (the current query functions are based around a template and episodes matching this template to some degree could be said to be an extended episode, but there is no native methods provided by episodic memory for handling these extended sets of episodes).
- Have some salience variable that is changed based on recency of recognition of different situations that is a function of the number of times a particular situation has been recognized and how recent those events are—this can represent a bias in the individual's recent experience (e.g., "Now that I know what a Maserati looks like, I see them everywhere").
- A measure of emotionality or novelty can also have an influence on this salience measure and can influence likelihood of future novel situations being perceived as examples of the highly emotional or salient one (cf. Schupp, Junghofer, Weike, and Hamm, 2003).
- Make population of the model more realistic in that instead of hard-coding situations and concepts into the model as a bootstrap into situation recognition, have the episodic memory feed information back into a more primitive semantic network to provide a richer, conceptualized understanding of the person's mental model.
- Include a discrepancy detection function in a model equipped with episodic memory such that memory for specific experiences informs discrepancy detection.
- Provide a way for "dormant" situations to drop out of the situation library when that salience value is low enough.
- Utilize the learning capabilities of episodic memory to aid in automated knowledge elicitation by enabling a user to customize a generic domain model through interaction with the model.
- Constrain episodic memory to some maximum size by pruning and develop the pruning methods that do not degrade the utility of episodic memory. The current model assumes that episodic memory is unconstrained.

4. Acknowledgements

This work was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

5. References

Allan, K., Robb, W.G.K. and Rugg, M.D. (2000). The effect of encoding manipulations on neural correlates of episodic retrieval. *Neuropsychologia*, *38*, 1188-1205.

Baguley, T. and Payne, S.J. (2001). Long-term memory for spatial and temporal mental models includes construction processes and model structure. *Quarterly Journal of Experimental Psychology*, *53A*(2), 479-512.

Bouton M.E., Nelson J.B. and Rosas J.M. (1999). Stimulus generalization, context change, and forgetting. *Psychological Bulletin*, *125*(2), 171-186.

Butterfield, E.C. and Nelson, G.D. (1989). Theory and practice of teaching for transfer. Educational Technology Research and Development, 37, 5-38.

Cantrambone, R., and Holyoak, K.J. (1989). Overcoming contextual limitations on problem solving transfer. Journal of Experimental Psychology: Learning, Memory, and Cognition, 15, 1147-1156.

Coles, M.G.H., Snid, H.G.O.M, Scheffers, M.K. and Otten, L.J. (1995). Mental chronometry and the study of human information processing. *Electrophysiology of mind: Event-related brain potentials and cognition*, Oxford: Oxford Science Publications, 86-131.

deWaal, F. (1989). *Chimpanzee Politics: Power and Sex Among Apes*, Baltimore: John Hopkins.

Dosher, B.A. and Rosedale, G. (1991). Judgments of semantic and episodic relatedness: Common time-course and failure of segregation. *Journal of Memory and Language*, *30*, 125-160.

Fletcher, P. C., Shallice, T.and Frith, C. D. (1996). Brain activity during memory retrieval: The influence of imagery and semantic cueing. *Brain*, *119*(5), 1587-1596.

Forsythe, C. (1999). *Theoretical Framework for Combatants for Next Generation Security Siulation: Human Factors Analysis*. Unpublished Manuscript.

Forsythe, C. (2001). *Comprehensive Framework for the Computational Representation of Human Decision Processes*. SAND 2001-1844, Sandia National Laboratories, Albuquerque, NM.

Forsythe, J.C., Bernard, M.L., Xavier, P., Abbott, R., Speed, A., and Brannon, N. (2002). Engineering a Transformation of Human-Machine Interaction to an Augmented Cognitive Relationship. SAND 2003-0043P, Sandia National Laboratories, Albuquerque, NM.

Hofstadter,D.R. (2001). Epilogue: Analogy as the core of cognition. In D.Gentner, K.J. Holyoak, and Kokinov, B.N. (Eds) The Analogical Mind: Perspectives from Cognitive Science. 499-538. Cambridge, MA: MIT Press.

Gentner, D. and Markman, A.B. (1997). Structure mapping in analogy and similarity. American Psychologist, 52, 45-56.

Gick, M.L. and Holyoak, K.J. (1983). Schema induction and analogical transfer. Cognitive Psychology, 15, 1-38.Goldsmith, T.E., Johnson, P.J. and Acton, W.H. (1991). Assessing structural knowledge. *Journal of Educational Psychology*, 83*(1)*, 88-96.

Gottlieb, E.J., McDonald, M.J., Oppel, F.J., Rigdon, J.B., and Xavier, P.G. (2002). "The Umbra Simulation Framework as Applied to Building HLA Federates," *Proceedings of the 2002 Winter Simulation Conference*. December 2002, San Diego, CA.

Holyoak, K.J. and Hummel,J.E. (2001). Toward an understanding of analogy within a biological symbol system. In D.Gentner, K.J. Holyoak, and Kokinov, B.N. (Eds) *The Analogical Mind: Perspectives from Cognitive Science.* 161-195. Cambridge, MA: MIT Press.

Jordan, S.E., Forsythe, C., Speed, A.E., Wenner, C., and Golsdmith, T.E. (2002). Extensibility of knowledge-based human agent simulation. SAND2002-3714, Sandia National Laboratories, Albuquerque, NM.

Kaufman, L., and Rousseeuw, P.J.(1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley.

Kerstholt, J.H. and Jackson, J.L. (1998). Judicial decision making: Order of evidence presentation and availability of background information. *Applied Cognitive Psychology*, *12*, 445-454.

Klein, G. (1997). An overview of naturalistic decision making applications. In C.E. Zsambok and G. Klein *Naturalistic Decision Making*, Mowah, NJ: Lawrence Earlbaum, 49-59.

Klein, S. B., Sherman, J. W. and Loftus, J. (1996). The role of episodic and semantic memory in the development of trait self-knowledge. *Social Cognition*, *14*(4), 277-291.

Klimesch, W., Schimke, H. and Schwaiger, J. (1994). Episodic and semantic memory: An analysis in the EEG theta and alpha band. *Electroencephalography and Clinical Neurophysiology*, *91*, 428-441.

Klimesch, W. (1996). Memory processes, brain oscillations and EEG synchronization. *International Journal of Psychophysiology*, *24*, 61-100.

Kok, A. (1990). Internal and external control: A two-factor model of amplitude change of event-related potentials. *Acta Psychologica*, *74*, 203-236.

Krause, W., Gibbons, H. and Schack, B. (1998). Concept activation and coordination of activation procedure require two different networks. *Cognitive Neuroscience*, *9*, 1649-1653.

Krause, K., Korpilahti, P., Porn, B., Jantti, J. and Lang, H.A. (1998). Automatic auditory word perception as measured by 40 Hz EEG responses. *Electroencephalography and Clinical Neurophysiology*, *107*, 84-87.

LeDoux, J. (1998). *The emotional brain: The mysterious underpinnings of emotional life*. Touchstone Books.

MacNeilage, P.F. (1998). The frame/content theory of evolution of speech production. *Behavioral and Brain Sciences*, *21*, 499-546.

Millis, K. K. and Cohen, R. (1994). Spatial representations and updating situation models. *Reading Research Quarterly*, *29*(4), 368-380.

Moll, M. and Milkkulainen, R. (1997). Convergence-zone episodic memory: Analysis and simulations. *Neural Networks*, *10*(6), 1017-1036.

Nyberg, L. and Tulving, E. (1996). Classifying human long-term memory: Evidence from converging dissociations. *European Journal of Cognitive Psychology*, *8*(2), 163-183.

Rugg, M.D. and Coles, M.G.H. (1995). *Electrophysiology of Mind: Event-Related Potentials and Cognition*. New York: Oxford University Press.

Schmidt, S.R. (1996). Category typicality effects in episodic memory: Testing models of distinctiveness. *Memory and Cognition*, *24*(5), 595-607.

Schupp, Junghofer, Weike, and Hamm (2003). Emotional facilitation of sensory processing in the visual cortex. *Psychological Science, 14,* 7-13.

Shastri, L. (2000). Types and quantifiers in SHRUTI : A connectionist model of rapid reasoning and relational processing. *Hybrid Neural Systems*, *1778*, 28-45.

Shastri, L. (2001). A computational model of episodic memory formation in the hippocampal system. *Neurocomputing*, *38*, 889-897.

Sherman, J.W. and Bessenoff, G.R. (1999). Stereotypes as source-monitoring cues: On the interaction between episodic and semantic memory. *Psychological Science*, *10*(2), 106-110.

Shum, M.S. (1998). The role of temporal landmarks in autobiographical memory processes. *Psychological Bulletin*, *124*(3), 423-442.

Taylor, H. A., Naylor, S. J. and Chechile, N. A. (1999). Goal-specific influences on the representation of spatial perspective. *Memory and Cognition*, *27*(2), 309-319.

Trafmow, D. and Wyer Jr., R.S. (1993). Cognitive representation of mundane social events. *Journal of Personality and Social Psychology*, *64*(3), 365-376.

Tulving, E. and Kroll, N. (1995). Novelty assessment in the brain and long-term memory storage. *Psychonomic Bulletin and Review*, *2*(3), 387-390.

Uttl, B. and Graf, P. (1996). Object orientation information in semantic and episodic memory. *Canadian Journal of Experiemental Psychology*, *50*(1), 87-103.

Wang, R.F. (1999). Representing a stable environment by egocentric updating and invariant representations. *Spatial Cognition and Computation*, *1*(4), 431-445.

Watkins, M.J. (1990). Mediationism and the obfuscation of memory. *American Psychologist, 45,* 328-335.

Wheeler, M.A., Stuss, D.T. and Tulving, E. (1997). Toward a theory of episodic memory. *Psychological Bulletin*, *121*(3), 331-354.

Wiggs, C.L., Weisberg, J. and Martin, A. (1998). Neural correlates of semantic and episodic memory retrieval. *Neuropsychologia*, *37*, 103-118.

Wilding, E.L. (2000). In what way does the parietal ERP old/new effect index recollection. *International Jorunal of Psychophysiology*, *35*, 81-87.

Zimmer, H.D., Helstrup, T. and Engelkamp, J. (2000). Pop-out into memory: A retrieval mechanism that is enhanced with the recall of subject-performed tasks. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *26*(3), 658-670.

Zwaan, R. and Radvansky, G.A. (1998). Situation models in language comprehension and memory. *Psychological Bulletin*, *123*(2), 162-185.

6. Appendices

6.1 Appendix A. List of Status Dimensions

Dimension	Description
1	Time elapsed following start of simulation
2	Current *X*-coordinate: X_t
3	Current *Y*-coordinate: Y_t
4	Absolute value of change in *X*-coordinate (since previous epoch): X_t - X_{t-1}
5	Absolute value of change in *Y*-coordinate (since previous epoch): Y_t - Y_{t-1}
6	Current smoke level: S_t
7	Change in smoke level since previous epoch: S_t - S_{t-1}
8	Current smoke relative to maximum smoke since start: S_t - max{ S_1, S_2, ..., S_t}
9	Current smoke relative to global (over all robots) maximum smoke since start
10	IS_Beacon: binary variable that indicates whether or not robot is a beacon
11	IS_Last: binary variable that indicates whether or not robot is last
12	IS_Rover: binary variable that indicates whether or not robot is a rover
13	RF_Hear_Beacon: binary variable that indicates whether or not robot can hear a beacon
14	RF_Ping: binary variable that indicates whether or not robot is pinging
15	STOP: binary variable that indicates whether or not robot is STOPed

6.2 Appendix B. Initial Situation and Concept Primitives

Situation	Concepts	Description
Seek Refuge	Dominant, Threat | Attack, Threaten | Attacker, Threatened | Attacked, Guardian, Seek Refuge, Provide Refuge	An entity is threatened by a dominant entity. A third entity known as a guardian is present. The entity receiving the threat seeks refuge from the guardian.
Recruit Support	Attack, Win, Lose, Attacker, Winner, Loser, Attacked, Ally, Recruit Support, Provide Support	An entity is attacked and defeated. There is a third entity known to be an ally by the entity defeated. The ally is approached and recruited to support action against the victor to achieve a reversal.
Instigate Attack	Adversary, Threat | Attack, Instigate, Instigator, Instigated, Threaten | Attacker, Threatened | Attacked	One entity is an adversary to another. To gain an advantage, the entity that is threatened provokes a third entity to either attack or threaten the first entity.
Reconciliation	Threat | Attack, Seek Reconciliation, Reconciliation, Accept Reconciliation	There is tension between two entities associated with recent threats or attacks. One entity seeks to reduce the tension through reconciliation.
Form Coalition	Adversary, Seek Coalition, Accept Coalition, Coalition	Two entities share a common adversary. The two entities form a coalition to resist or challenge this adversary.
Form Collaboration	Barrier, Seek Collaboration, Accept Collaboration, Collaboration	Same as Form Coalition, except that the two entities face the same barrier or blocked goal, as opposed to a common foe. The two entities collaborate to overcome the barrier enabling them to pursue their separate goals.
Desert Coalition	Adversary, Coalition, Desert Coalition	The entity for whom a coalition formed to oppose is not perceived to be an adversary by one or more members of the coalition. One or more members of the coalition desert.
Prohibit Desertion	Adversary, Coalition, Desert Coalition, Prohibit Desertion	One entity seeks to desert a coalition, however the original adversary remains a threat for the other member of the coalition. This member seeks to prohibit the desertion.

Interference	Threat | Attack, Ally, Adversary, Interference, Interfere, Interfered	There is a confrontation between two entities. The potential exists that a third entity may enter the confrontation due to allegiance to one of the adversaries. A fourth entity, due to either their allegiance or adversarial relationship with the original pair, prevents the third entity from joining the confrontation.
Arbitration	Threat | Attack, Ally, Arbitration, Arbitrator, Arbitrated	There is a confrontation between two entities. A third entity that is perceived to be relatively neutral arbitrates leading to reduction in tension between the two adversaries.
Evoke Sympathy Reduce Threat	Threat | Attack, Threaten | Attacker, Threatened | Attacked, Evoke Sympathy, Sympathy, Sympathize	Entity threatens or attacks a second entity. The second entity exhibits a display of weakness seeking to elicit a reduction in the threat.
Evoke Sympathy Mobilize Support	Threat | Attack, Threaten | Attacker, Threatened | Attacked, Evoke Sympathy, Sympathy, Sympathize, Provide Support	Entity threatens or attacks a second entity. The second entity exhibits a display of weakness seeking to elicit support from a third entity.
Enforced Silence	Adversary, Enforced Silence, Silence Enforcer, Silence Enforced, Instigation, Instigator, Instigated	Either the situation is appropriate for instigation or instigation has occurred. By enforcing silence, the entity that would be the target of an instigated attack seeks to terminate or prevent instigation.
Provocation Avoidance	Threat | Attack, Threatened | Attacked, Threaten | Attacker, Conceal Signal, Concealed Signal, Signal	Situation exists in which a threat or attack is probable. One entity seeks to conceal any signal that could provoke a threat or attack. The resulting behavior may include hiding.
Display Dominance	Dominance, Threat | Attack, Threatened | Attacked, Threaten | Attacker	One entity displays through either a threat or attack as a means of sustaining dominance relative to other entities.
Displacement	Dominance, Resource, Acquire Resource | Possess Resource, Drive Resource, Displacer, Displacement, Displaced	One entity either possesses a resource or is in the process of acquiring the resource. A second entity has a drive for the same resource. Through a show of dominance, the second entity displaces the first.
Recognize Weakness	Adversary, Threat | Attack, Threatened | Attacked, Threaten | Attacker, Injured, Fatigued, Support, Coalition	One entity recognizes that an adversary is injured or fatigued, or has lost support or a coalition. Through threats or attacks, the entity seeks to take advantage of weakness on the part of its adversary.
Regulate	Threat | Attack, Regulator, Regulate, Regulated	Tension and disorder results from the threats and attacks between two entities. A third entity intercedes to restore order.
Feign Good Intentions	Coalition | Collaboration, Adversary, Feign Good Intentions	Shared foe or blocked goal creates conditions for coalition or collaboration, however there is an adversarial relationship between the participants. By feigning good intentions, the effect of the adversarial relationship is diminished.

Recruit Support	Attack, Win, Lose, Attacker | Attacked, Winner, Loser, Ally, Seek Solace, Provide Solace	Occurs as an alternative to Recruit Support and may precede or accompany Recruit Support. An entity is attacked and defeated. There is a third entity known to be an ally by the entity defeated. Solace is sought from the ally as a means to restore loses accrued by defeated entity (e.g., dominance, fear reduction).
Pay Homage	Dominance, Pay Homage, Homage, Receive Homage	One entity acknowledges dominance of second entity as a means to avoid or delay displays of dominance
Mock Fight	Adversary, Play Threatener | Play Attacker, Play Threatened | Play Attack, Mock Fight, Dominance	Two entities are in a nonadversarial context. One entity initiates play threats or attacks with the intent that the second entity will respond with a play response. The activity may result in a nonthreatening adjustment to relative dominance relationships.
Overbearing Dominance	Adversary, Dominance, Seek Coalition, Coalition, Accept Coalition	Situation exists in which displays of dominance are of sufficient severity to prompt entities to unite in their resistance.
Forced Coalition	Adversary, Seek Coalition, Coalition, Accept Coalition, Threaten | Attacker, Threat | Attack, Threatened | Attacked	One entity seeks a coalition with a second entity to oppose an adversary of the first entity. The second entity does not accept the coalition. Consequently, the first entity employs threats in an attempt to force the second entity to form a coalition.
Undermine Coalition	Adversary, Coalition, Seek Reconciliation, Reconciliation, Accept Reconciliation	A coalition exists between two entities to oppose a third entity. The third entity seeks a reconciliation with one of the coalition partners in an attempt to break the coalition.
Keeping Up Appearances	Adversary, Injured | Fatigued | Support | Coalition, Conceal Signal, Signal, Concealed Signal	One entity is weakened due to injury, fatigue, or loss of support or a coalition. Actions are taken to conceal indications of this condition from the adversaries of the entity.
Assuage Temper Tantrum	Ally, Loser | Blocked Goal, Temper Tantrum, Assuager, Assuaged, Assuage Frustration	Due to having lost a confrontation or having experienced a blocked goal, one entity experiences frustration leading to a temper tantrum. A second entity that is an ally of the first seeks to assuage the frustration of the first entity.
Avoid Temper Tantrum	Loser | Blocked Goal, Temper Tantrum, Conceal Signal, Signal, Concealed Signal	Due to having lost a confrontation or having experienced a blocked goal, one entity experiences frustration leading to a temper tantrum. A second entity seeks to avoid provoking the first entity with hiding being a potential action.
Resist Challenge	Threatener | Attacker, Threat | Attack, Threatened | Attacked	Following a challenge, entity responds with threat or attack directed at challenger.
Succumb Challenge	Threatener | Attacker, Threat | Attack, Threatened | Attacked, Succumber, Succumb, Succumbed	Following a challenge, entity succumbs to the challenger.

Bluff Over	Bluff Threat | Attack, Threat | Attack, Threatened | Attacked, Dominant	To effect their relative status, one entity bluffs a threat or attack against a dominant entity such that other entities perceive the action to be a genuine threat or attack, however the threat or attack is withdrawn before eliciting a response from the dominant entity.
Blackmail	Requester, Request Action, Requested, Demander, Demand, Demanded	One entity requests action from a second. The second entity demands payment from the first with the promise of performing the requested action.
Subservient Advisor	Adversary, Dominance, Subserver, Subserved, Subservience	One entity seeks advantage over a third entity by assuming the role of a subservient to a more dominant entity.
Defend Weak	Threatener | Attacker, Threat | Attack, Threatened | Attacked, Provide Support, Support	A stronger entity threatens or attacks a weaker entity. A third even stronger entity takes action to defend the weaker as a means of building support for their dominance.
Demonstration Unity	Threat | Attack, Coalition, Demonstrate Unity, Demonstration Unity	Two entities are partners in a coalition. One of the coalition partners is the subject of a threat or attack. There is a demonstration of unity by the coalition partners to deter continued or future threats or attacks.
Systematic Reprisal	Dominance, Opposition, Threatener | Attacker, Threat | Attack, Threatened | Attacked	A dominant entity is subject to opposition from a second entity. The dominant entity employs threats or attacks as a means to deter continued or future opposition.
Shared Leadership	Dominant, Offer Leadership, Accept Leadership, Shared Leadership	A dominant entity seeks to deter challenges from a second entity of near equal strength by offering to share leadership with the second entity.
Non-Intervention	Adversary, Offer Non-Intervention, Non-Intervention, Accept Non-Intervention	Two entities share a common adversary. The two entities agree to not intervene on behalf of their common adversary in a dispute involving one of the two entities.
Distribution of Wealth	Possess Resource, Desire Resource, Offer Resource, Accept Resource, Distribution Resource	One entity possesses a resource that is desired by a second entity. As a measure to enhance status, the first entity gives the second entity some of the desired resource.
Begging for a Share	Possess Resource, Desire Resource, Begger, Begging, Begged	One entity desires a resource possessed by a second entity. The first entity begs in attempt to get the second entity to give them some of the resource.

Distribution

1	MS 9201	Howard Hirano (8101)
1	MS 0318	David Schoenwald (9216)
1	MS 0829	Edward Thomas (12323)
1	MS 0830	Kathleen Diegert (12335)
1	MS 0830	Ann Speed (12335)
1	MS 1004	Patrick Xavier (15221)
1	MS 1170	Russel Skocypec (15310)
1	MS 1188	John Wagner (15311)
1	MS 1188	J. Chris Forsythe (15311)
1	MS 1188	Carl Lippitt (15311)
1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library
1	MS 0612	Review and Approval Desk, 9612
1	MS 0188	LDRD office
5	MS1188	Department File (15311)