A COGNITIVE AGENT ARCHITECTURE OPTIMIZED FOR ADAPTIVITY

ANNA L. BUCZAK
Sarnoff Corp., 201 Washington Avenue, Princeton, NJ 08543-5300

KSHANTI GREENE, DAVID G. COOPER, MICHAEL CZAJKOWSKI, MARTIN O. HOFMANN
Lockheed Martin Advanced Technology Laboratories, 3 Executive Campus,
Cherry Hill, NJ 08002

ABSTRACT

The Engine for Composable Logical Agents with Intuitive Reorganization (ECLAIR) is a cognitive agent architecture that allows an agent to quickly adapt its behavior to new environments. ECLAIR addresses two problems in agent learning: generalizing the process of adaptation and detecting when adaptation is required. ECLAIR incorporates the main mechanisms from Piaget's Cognitive-Stage Theory of Development [10], and it uses concepts from Damasio's Somatic Marker Hypothesis [4] for discovery of what should be learned. ECLAIR has modules for stimuli, awareness, plan behavior, reflex behavior, control/decision making, and adaptivity. ECLAIR agents take a hybrid approach to action. In normal situations they act logically, using plans, when there is a known strategy to accomplish a task. However, when quick reaction is needed, motivation for action is intuitive or emotional. The agent fires reflexes triggered by changes in the agent's perception of personal well being. Adaptive learning extends both cognitive and emotional behavior in our architecture. We demonstrated the cognitive architecture and reflexive adaptation using a simulation for network-centric warfare logistics.

1. COGNITIVE ARCHITECTURES

Researchers have strived to develop cognitive architectures since the 1980s. The main premise of these types of architectures is to emulate the human, how a person makes decisions, represents information, and learns. The two most widely known cognitive architectures with a psychological basis are SOAR [9] and ACT-R [1]. Both are hypotheses for answering Newell's concept of a United Theory of Cognition [12]. Newell saw that in a person, there are many interacting components that must be integrated into a single comprehensive system, and believed that the single system is the source of all behavior. Thus, the goal of a cognitive architecture is to have one system that gives purpose to the many components that make up a thinking person.

ACT-R is a cognitive architecture designed as an integration of components discovered in psychology research. This model is primarily meant to accurately simulate human behavior. Given a specific cognitive theory, ACT-R can be used to model the components of the theory. Once the model has been created, experiments can be made that get results very similar to human experimental results. In addition, the model can be used to extend previous theories by creating a novel experiment for the model. ACT-R also has a set of modules that represent different functional aspects of the brain. The modules interact when each module exposes part of its activity into a public buffer. The central production system uses the data in the buffers for its processing. ACT-R has primarily been used for psychological research [1].

SOAR is a cognitive architecture focused on the functional requirements of human level intelligence. SOAR attempts to recreate goal driven behavior, continuous learning from experience, and real time cognition. The goal is to have a system where memory can be directly used for action. A production system is at the heart of the architecture. The decision cycle has seven steps: Input, State Elaboration, Propose Operator, Compare Operators, Select Operator, Apply Operator, and Output [11]. SOAR's mechanism for learning, called "chunking," has proven to cause unexpected results and experiments have been done to replace the chunking mechanism by reinforcement learning [11].

The paper is organized as follows: Section 2 describes the theoretical background; Section 3 is the architecture description; Section 4 talks about adaptivity in detail; Section 5 describes the logistics application, and Section 6 concludes.

2. MOTIVATION AND THEORETICAL BACKGROUND FOR ECLAIR

The Engine for Composable Logical Agents with Intuitive Reorganization (ECLAIR) was designed as a cognitive agent architecture amenable to adaptivity and learning. Agent adaptivity and learning are important for real world domains, where the situation in the environment dynamically changes and the agents have to adapt quickly. One domain of special interest is Sense and Respond Logistics (SRL). SRL aims to deliver precise, agile support through adaptive and responsive demand and support networks [13]. Automation technologies supporting SRL should respond to events that occur, as well as aid in the perception and anticipation of short-term needs. The cognitive agents working in NCL domains must adapt by reorganizing themselves to suit the environment, or by modifying their behavior to improve their effectiveness.

ECLAIR is a cognitive model based on developmental cognitive psychology research and neuropsychological research. Though many developmental theories contributed to the ECLAIR model, the two most prominent are Piaget's adaptation theory [10], and Damasio's Somatic Marker Hypothesis [4]. Piaget's adaptation theory consists of three main concepts: *Assimilation*, *Accommodation*, and *Equilibration*. *Assimilation* processes unfamiliar input in the same way that one would process the most similar familiar input. *Accommodation* changes the processes to deal with unfamiliar input. *Equilibration* balances the aforementioned processes. The *Somatic Marker Hypothesis* stems from Damasio's belief that gut feelings are a basis for decisions in addition to reasoning, which is often thought of as the only basis for decision. Damasio defines a somatic marker as a trigger that recalls feelings related to previously made decisions. The decision is made based on the best expected feeling given the available actions for the current circumstance. Each memory of feelings becomes a somatic marker that is used as a mapping from circumstance to action.

ECLAIR agents contain modules for stimuli, awareness, plan behavior, reflex behavior, control/decision making, and adaptivity. The interaction between awareness, behavior and adaptivity allows agents to modify their behavior based on their perception of world and self-states. Self-states are represented by homeostatic vectors (HVs), where the comfortable level for the agent is a range, not a threshold. Agent well being is an emotional state that is computed as a function of the agent's HVs.

3. ECLAIR ARCHITECTURE

The main elements of ECLAIR architecture (Fig. 1) include the following modules: Stimulus, Awareness, Reflex Behavior, Plan Behavior, Adaptivity, and Action/Decision.

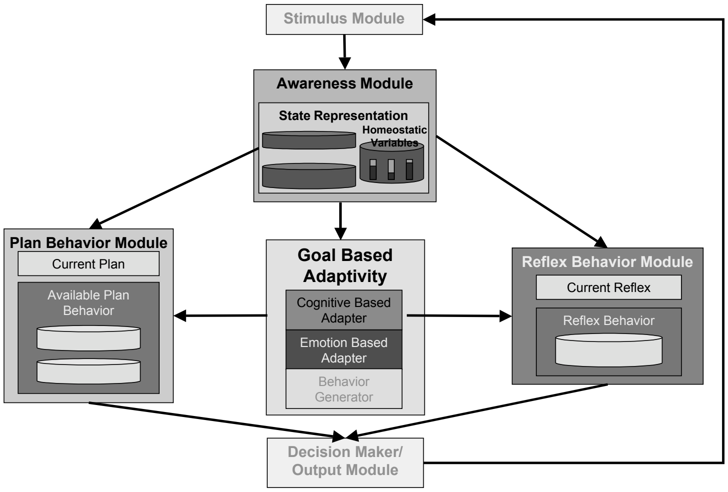

Fig. 1. ECLAIR Architecture.

The Stimulus Module deals with the perception of the world by the agent. There are two types of perception: external and internal. For a living being, external perception is in the form of sensed sound, smell, sight, taste, and touch, while internal perception may include hunger, pain, and comfort. The Stimulus Module tries to match the input (perception) to a known stimulus. If it is successful, the agent has dealt with this type of input previously, and knows how to respond (has an appropriate plan or reflex). If the input remains unmatched to a known stimulus, the agent does not know how to respond, as it did not previously encounter and suitably solve a similar problem. In this case, depending on the Awareness Module output, the agent will have a choice of performing assimilation or accommodation (described in detail in the Adaptivity Module).

The Awareness Module maintains a representation of what the agent is aware of in the world and of self. It performs: 1) maintenance of the state of the world and self, based on inputs; 2) maintenance of the predicted state of the world and self, based on the current state of the world, and the actions taken by self; and 3) mapping of expectations to observations. The awareness module creates an expectation when an action is taken, indicating what it expects to happen as a result of the action. Observations are stimuli that have been caused by an agent's previous action. Each expectation has a corresponding observation that indicates the actual result. Disconnects between expectation and observation may trigger behavioral adaptation in the form of accommodation.

Homeostatic vectors, representing current levels for different features of the agent, are part of the agent's self-state. With each HV there is a range of values associated representing the ideal values for each of the HVs. As each HV departs from its ideal range, the agent's overall *well being* decreases. *Well being* is an aggregate value that describes the agent's happiness. When the value of *well being* is low, the agent, similarly to humans, is frustrated. In a logistic application, HVs can be related to the agent's levels of fuel, ammunition, food, etc.

There are two types of behaviors that an ECLAIR agent can perform: a reflex or a plan. The Reflex Behavior Module is responsible for the first type of behavior, while the Plan Behavior Module is responsible for the second type. Agent reflexes are similar to human reflexes: they are simple actions that the agent performs. An example of a human reflex is to blink the eye when bright light suddenly appears. In case of a military logistic application, an agent's reflex can be to ask for ammunition when the agent has little left. Agent reflexes, like human reflexes, can be inhibited. Inhibition is handled by the Decision Module discussed later. Each reflex is registered with the Awareness Module. When the right stimulus comes in (i.e. matches the stimulus needed to fire a given reflex) it is passed to the Reflex Module, and the Reflex Module fires the reflex if it has not been inhibited. In addition, once a reflex is fired, it inhibits itself for a certain period of time so that the effect of the reflex can be observed.

The Plan Behavior Module is responsible for agent plans (also called workflows). A workflow is a list of tasks linked by conditional or unconditional paths. Tasks on an unconditional path are always executed, while tasks on a conditional path are only executed when the condition is true. An example of agent's workflow is shown in Fig. 2. The Plan Behavior Module can have multiple workflows available to the agent. With multiple workflows an agent can choose a different plan when the current workflow is at an impasse. If there is a higher priority workflow, it can interrupt the current workflow.

Both reflex and plan behaviors have conditions that need to be fulfilled for the behavior to be initiated. These conditions are in the form of a stimulus. When the Awareness Module receives a stimulus, it matches the stimulus with appropriate reflex and plan behaviors that have the stimulus type as a condition (e.g. a feeling of hunger may initiate a search for food). In addition, a stimulus may update the self and world representations (e.g. bumping into a wall may update the world representation that previously indicated a clear path).

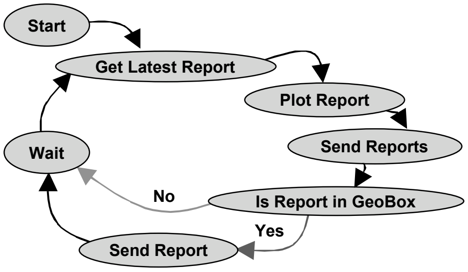

Fig. 2. Example of an ECLAIR agent's plan (workflow).

The Decision Module decides upon the agent's next behavior and allows for some flexibility of the agent. As mentioned previously, the behavior is either handled by a reflex, or by part of a plan. If a reflex is fired, the activity within it will be completed. If a plan is enacted, the plan will continue unless a higher priority plan is started. Plans result in execution of tasks that can contain actions. For a human, examples of action are speaking, moving, or manipulating. For agents, actions may be sending data, retrieving data, computing, sending control commands, ordering supplies, etc. Reflex inhibitions are handled in the Decision Module: an inhibition is created when there is a need for a certain action to be suppressed. The inhibition is removed once this requirement no longer exists. An example is an agent firing a reflex to order food whenever it is low on food. The reflex must be inhibited for a certain time after it is fired, or the agent would constantly fire the same reflex, resulting in ordering huge amounts of food.

The Adaptivity Module is where all of the adaptation/learning takes place. An agent can be created without the adaptivity module. This agent would act predictably, but would not adapt even if its actions failed to accomplish the desired task. This type of agent would have only a choice of predefined reflexes and plans, as well as predefined conditions indicating when to enact a given reflex or plan. Most deployed agent systems lack the adaptivity/learning capability.

4. ADAPTIVITY IN ECLAIR

4.1 INTRODUCTION

Goal Based Adaptivity happens in parallel to the other agent activities. It allows the agent to respond quickly to what is happening in the environment (i.e. make fast decisions) but at the same time to learn and improve its behavior. There are three types of adaptivity in our agents: Emotion-Based, Cognitive-Based, and a New Behavior Generator. Emotion-Based Adaptivity performs short term learning and is triggered by the agent's emotions, such as frustration. Emotional actions are less explicative than cognitive ones, but they can extend prediction with innate sensibility and they have faster reaction time [5]. This type of adaptivity is fully implemented in our system and is described in Section 4.2. Cognitive Based Adaptivity performs long term learning and the agent uses reasoning to learn and improve its behavior. The agent reasons on many levels, and uses explicit rule knowledge. Cognitive adaptivity makes more accurate predictions based on rules of causality and complements the Emotion-Based Adaptivity similarly to Asynchronous Learning by Emotion and Cognition system [5]. Cognitive adaptivity is currently a placeholder in our architecture that has not been implemented yet. The New Behavior Generator is a mechanism that causes a completely new behavior to be learned: this can be a reflex or a plan. The mechanism that we envision for this purpose is Genetic Programming- the same methodology that is described in Section 4.3 for Plan Adaptivity. Again, this part awaits a future implementation.

4.2 EMOTION-BASED ADAPTIVITY

Emotion-Based Adaptivity is triggered by the "emotions" of the agent. In nature, emotions are pleasant or unpleasant feelings that interrupt current behavior. In ECLAIR the short cut for emotions is the agent *well being*. When the agent's *well being* decreases (as determined by the Awareness Module), the agent becomes frustrated, similarly to what happens in people. For example, this could occur if an agent experiences a stimulus with no corresponding response. In this case, the stimulus is unmatched and the agent has a choice of performing assimilation or accommodation. Assimilation is performed when the current stimulus is similar to another stimulus for which the agent has a previously used (known) response, and the known response is executed. Accommodation can take place when the current stimulus is very different from stimuli that the agent knows how to respond to. Accommodation implies that the agent has to modify a known behavior, and perform this new behavior to suit the new stimulus. The successful accommodation will be recognized by the agent through its improved *well being.* The structure of the Goal Based Adaptivity Module allows for different learning mechanisms to be used. The plan adaptivity and reflex adaptivity described below are examples of learning mechanisms. Both mechanisms perform accommodation, i.e., change a known agent behavior into a new, better behavior.

4.3 PLAN ADAPTIVITY

The plan adaptivity mechanism was designed for agents with a workflow similar to Fig. 2. Our approach to plan adaptivity is named Evolutionary Platform for Agent Learning (EPAL) [3]. Genetic Programming (GP) invented by John Koza [8] constitutes the basis for adaptivity in EPAL. Using the principles of Darwinian evolution, GP performs program synthesis by genetically breeding a population of computer programs. The basic operators of reproduction, crossover and mutation operate on individuals in the population and a fitness function describes the goodness of an individual. In GP each individual program is represented as a tree.

In EPAL we represent agent plans in a GP tree form and GP operators work on agents' genetic material (i.e., GP trees) to generate new agents that have learned to overcome certain problems in their environment. As agents execute in the environment their fitness is collected. The value of fitness guides the evolutionary learning process. This method is a general method that can generate completely new agent plans as well as related plans, but with new parameters. Augmenting an agent's plan is synonymous with changing the agent's behavior, thus the method can be used for generating new behaviors in addition to modifying parameters. Our ECLAIR software agents are more complicated than the small software programs that GP usually evolves. In order to evolve meaningful agents in a realistic time frame our representation of GP agents needs to be at a higher level than simple Java instructions and their parameters. EPAL's main GP building blocks are the individual tasks that compose a workflow. We have not used the EPAL agent plan adaptivity in a logistics scenario, although we are currently integrating EPAL into ECLAIR's plan adaptivity module. The interested reader can find the details of GP-based adaptivity in [2] and [3].

4.4 REFLEX ADAPTIVITY

A reflex in ECLAIR is composed of a stimulus, an activity, and a set of parameters for the activity. We use an approach based on Reinforcement Learning (RL) [7, 13] to learn the best parameters to use in an activity given the stimulus. RL is based on two major principles: receiving immediate reinforcement for taking actions in an environment given the state of the environment, and generating an overall value for a state-action mapping using delayed reward. Our reinforcement approach calculates the overall values of stimulus-activity-parameter mappings from the reward received as the results of activities are observed. The innovation in our technique is not the technical aspects of our RL algorithm, but the use of cognitive elements from the architecture, such as perception and expectation, as the foundation for reward.

A typical RL problem is composed of a set of discrete states and a set of discrete actions. The high-level goal is to learn the best mapping between state and action, or the best *policy*. In ECLAIR, a state is composed of a stimulus and an activity. Stimuli in the logistics scenarios include internal states (represented as HVs) such as Low_Fuel or Very_Low_Food. Given these internal states, our agents will do an activity; for example, Order_Fuel and Order_Food, respectively. The parameters of the activity are variables and constitute our learning problem. Parameters include who to order from, how much to order, and the order priority. Our policies are composed of {stimulus-activity}-{parameter set} mappings, corresponding to RL's state-action mappings. The reinforcement learner determines which parameter set to use given a stimulus-activity pair. Reinforcement is computed by comparing the expectation of the activity with the observed results. This was a significant challenge in our approach as the observations from an activity are not immediate and may not be seen until several intermediate tasks are completed. In our logistics simulation, the expectation from Order_Fuel is that an agent will receive the amount of Fuel it ordered *within a certain period of time*. In order for an agent to receive a resource, a supplying agent must receive the order, and then must send out an asset to complete the order, assuming it has the asset and resource available. The whole operation could potentially take several simulated days, even with a relatively fast chain of command. If the supplying agent does not have an available asset or the requested resource, the order may never be filled.

Overall value for a policy is the summation of its reinforcement over time. When selecting a policy to use for a given state, usually the policy with the highest overall value is used. However, exploration will occur at a rate dependent on the *well being* of the agent. If *well being* is high, then the agent is doing well with the default or learned behaviors and will decrease the exploration rate. However, if *well being* is low, then the agent will increase the rate so that the agent explores more often and perhaps discovers better policies to improve *well being.* Balancing exploration with exploitation is a difficult problem in RL [14]. Our solution incorporates knowledge about the internal emotional state to compute an exploration rate that is well suited at a given moment.

5. LOGISTICS APPLICATION

5.1 SIMULATION

We created a demonstration that shows how ECLAIR agent adaptivity works in a Sense and Respond Logistics (SRL) scenario. The scenario demonstrates ECLAIR's ability to react appropriately in a dynamic environment. In our scenario, each operational unit (OU) and each supply unit (SU) is represented by an ECLAIR agent. As an OU, the ECLAIR agent monitors its homeostatic vectors that indicate fuel, ammunition, and food amounts. As the OU consumes its resources, it becomes increasingly unhappy (low *well being*) and it requests a re-supply from a particular SU. OUs set their expectations based on how much of a resource they requested and how long they expect to wait for the request to be fulfilled. When supplied, the OU agent observes how much of the resource it received and how long it had to wait. The expectations and observations of an OU influence its decision to continue using a particular SU or to choose a new one.

We tested ECLAIR agents in a scenario with three OUs and two SUs. One supply unit, *SU1* has nine assets (i.e. helicopters or trucks to transport supplies) and the other, *SU2* has only one. *OU1*, *OU2*, and *OU3* are initially assigned a default supply unit to order resources from: OU1 requests resources from SU1; OU2 and OU3 request resources from SU2. The demonstration uses the concept of Operational Availability (Ao) as a metric in determining whether adaptation is truly occurring.

Operational Availability is a metric used by the military to determine how prepared a unit is to handle its commands. In our demonstration prototype, we model Ao as a function of a unit's outstanding resource needs, assuming that for a logistics scenario, the factor of interest is a unit's resource level. Ao can be thought of as a military version of well being, however in our scenarios, the Ao score was used as an externally observed evaluation of the scenario whereas well being was used internally by the agents to evaluate their own behavior.

Ao is modeled as a function: Ao = $MAX\text{-}sum(Pi|i = 1..n),$ where MAX is a perfect Ao score, i is a particular unit, n is the number of units, and Pi is the Personal Ao (pAo) of each operating unit. The pAo of each operating unit is calculated on the basis of outstanding supply requests by each unit i. pAo is modeled as a function $Pi = sum(Rj|j = 1..n)$ where Rj is the value of an outstanding order request j, and n is the number of requests outstanding. *Rj* is determined based on the state of the operational unit *i*: (engaged, moving, or idle) as well as the type of request made (fuel, ammunition, food), and how long ago the request was made. Certain states and request types have a higher priority than others and yield a higher *R* value. Particularly, a request for fuel while the unit is engaged yields the highest value. *R* will also increase over time at a different rate for each request type, indicating that as time passes it becomes more of a concern to be low on higher priority resources.

5.2 RESULTS

In our scenario, SU2 was "handicapped" with only one asset, compared to SU1's nine assets. In order to add an additional element of instability in the environment, pop-up enemy units periodically attacked OU3, causing a sudden increase in the need for ammunition. In most runs of this scenario, Ao initially decreased rapidly until the ECLAIR agents learned to choose different supply units based on the availability of resources. Within a short period, OU2 and OU3 learned to decrease the expectations of SU2's reliability because it was not delivering resources at a satisfactory level. Also, as *well being* decreased, more exploration occurred, causing the OUs to send their requests to SU1. Eventually, OUs almost always requested from the SUs that supplied the right types of resources in the shortest period of time.

Fig. 3 shows the average Ao score of 30 runs of our scenario for three settings *Default*, *Explore*, and *Adapt*. The dark gray line, marked "Default,"' shows the results of agents only requesting supplies from their default supply units i.e. agents that do not adapt. The black line, marked "Explore," shows the results of agents selecting a random supply unit whenever their *well being* is low, but not using the results to learn from whom to request supplies. The white line, marked "Adapt," shows the results of agents using reflex adaptivity i.e., learning to request supplies from the SU that provided them in a satisfactory fashion.

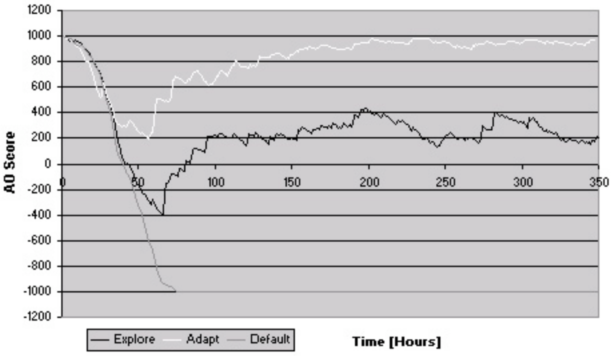

Fig. 3. Average Ao scores for 30 runs, over a period of 350 simulated hours.

Our results indicate that learning to adapt to the environment yields a clear improvement in Ao. The need for fuel, ammunition, and food varied from hour to hour in all scenarios. At the end of the Adapt scenario, OU2 and OU3 were being supplied faster by asking the more responsive SU1 for supplies. The Ao was improved from its initial setting when SU2 was supplying for all of the needs both OU2 and OU3. Additional scenarios that show adaptation to several dynamic changes in the environment are described in [6].

6. CONCLUSIONS

ECLAIR, the agent architecture we developed, realizes a cognitive model based on Piaget's Cognitive-Stage Theory of Development and Damasio's Somatic Marker Hypothesis. Agents take a hybrid approach to action, using plan-based behavior in normal situations, and emotionally inspired reflex behavior when they perceive internal distress. Adaptivity can manipulate plan and reflex behavior, improving agents' performance and adapting their response to changes in their environment. Using Goal Based Adaptivity, the agents learn how to better operate in a dynamic environment.

In the logistics scenario where we tested the ECLAIR agents, we demonstrated that adaptive agents greatly improve behavior over agents that do not adapt. In our demonstration, we show that the ECLAIR agent framework provides solutions to SRL challenge problems; agents improve the Ao score and adapt to the changes in a demand driven network. Most importantly, the adaptivity that enables this improvement is task and domain independent.

7. REFERENCES

[1] Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., Qin, Y. "An integrated theory of mind", *Psychological Review* Vol. 111, No. 4 (2004), pp.1036-1060.

[2] Buczak, A.L., Cooper D.G., Hofmann, M.O., "Advances in Evolutionary Agent Learning", *Intelligent Engineering Systems Through Artificial Neural Networks*, eds. C.H. Dagli, A.L. Buczak, J. Ghosh, M.J. Embrechts, O. Ersoy, Vol. 13, (2004), pp. 201-206, ASME Press, NY.

[3] Buczak, A.L., Cooper, D.G., Hofmann, M.O. "Evolutionary Platform for Agent Learning," *Intelligent Engineering Systems Through Artificial Neural Networks*, eds. C.H. Dagli, A.L. Buczak, D.L. Enke, M.J. Embrechts, O. Ersoy, Vol. 14, (2003), pp. 157-164, ASME Press, NY.

[4] Damasio, A.R. *Descartes' Error: Emotion, Reason, and the Human Brai,.* G.P. Putnam, NY (1994).

[5] Gadanho, S.C. "Learning Behavior-Selection by Emotions and Cognition in a Multi-Goal Robot Task," *Journal of Machine Learning Research*, Vol. 4, (2003), pp. 385-412.

[6] Greene, K., Cooper, D.G., Buczak, A.L, Czajkowski, M., Vagle, J.L, Hofmann, M.O., "Cognitive Agents for Sense and Respond Logistics," *Pre-proceedings of the Workshop on Defence Applications & Multi-Agent Systems held at the Fourth International Conference on Autonomous Agents and Multi-Agent Systems*, eds. R. Ghanea-Hercock, M. Greaves, N. Jennings, S. Thompson, (2005), pp 79-93.

[7] Kaelbling, L.P., Littman, M.L., and Moore, A.W. "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, Vol. 4, (1996), pp. 237-285.

[8] Koza, J., "Genetic Programming – On the Programming of Computers by Means of Natural Selection," Cambridge, Massachusetts, MIT Press (1993).

[9] Lewis, R.L., *Cognitive theory, SOAR.*, Tech. Rep., Ohio State University (1999).

[10] Miller, P.H., *Theories of Development Psychology*. W.H. Freeman and Co. (1983).

[11] Ason, S., and Laird, J.E. *Soar-Rl: Integrating Reinforcement Learning with Soar*. Tech. rep., University of Michigan (2004).

[12] Newell, A. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, (1990).

[13] Office of Force Transformation, 2004, United States Department of Defense, "Operational Sense and Respond Logistics: Coevolution of an Adaptive Enterprise Capability."

[14] Sutton, R.S., Barto, A.G., *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA (1998).