Cognitive Agents for Sense and Respond Logistics

Kshanti Greene[1], David G. Cooper[1], Anna L. Buczak[2]*,
Michael Czajkowski[1], Jeffrey L. Vagle[1], Martin O. Hofmann[1]

[1]Lockheed Martin
Advanced Technology Laboratories
3 Executive Campus, 6^{th} Floor
Cherry Hill, NJ 08002

[2]Sarnoff Corporation
CN 5300
Princeton, NJ 08543-5300

{kgreene, dcooper, mczajkow, jvagle, mhofmann}@atl.lmco.com, abuczak@sarnoff.com

Abstract

We present a novel cognitive agent architecture and demonstrate its effectiveness in the Sense and Respond Logistics (SRL) domain. Effective applications to support SRL must anticipate and adapt to emerging situations and other dynamic military operations. SRL transforms the static, hierarchical architectures of traditional military models into re-configurable networks designed to encourage coordination among small peer units. Multi-agent systems are ideal for SRL because they can provide valuable automation and decision support from low-level control to high-level information synchronization. In particular, agents can be aware of and adapt to changes in the environment that may affect control and decision making. Our architecture, the Engine for Composable Logical Agents with Intuitive Reorganization (ECLAIR) is based on cognitive theories for motivation and adaptation [6, 13, 21]. Agents respond to external stimuli and internal perception of wellbeing. In normal situations they act logically, using plans, or workflows, when there is a known strategy to accomplish a task. However, when quick reaction is needed, motivation for action is intuitive or reflexive. Adaptation using machine learning techniques improves both logical and reflexive behaviors in ECLAIR. To demonstrate and evaluate our approach, we implemented a small simulation environment where our agents handle the ordering and delivery of supplies among operational and supply units in several scenarios requiring adaptation of default behavior.

1. Introduction

Throughout history, smart warfighters and commanders have tried to deceive and confuse their opponents using any technological means at their disposal. Nation states no longer maintain a monopoly on armed forces, and this is fostering the transition to the next stage in the evolution of conflict: fourth generation warfare (4GW). The technologies developed to aid the warfighter in these emerging environments must be designed to support dynamic, adaptive operations. Sense and Respond Logistics (SRL) aims to deliver precise, agile support through adaptive and responsive demand and support networks [24]. Automation technologies supporting SRL should respond to events that occur, as well as aid in the perception and anticipation of short-term needs.

Quick, adaptive response requires small units at the subbattalion levels of the military hierarchy to be both more autonomous in their control, and more coordinated in their actions. Multi-agent systems can represent the varied roles of specific units and assets involved in logistics. They can automate some behaviors such as ordering supplies and prioritizing requests, and they can build an awareness of the world and other agents that allows them to enhance the decision making of unit commanders. Most importantly, automated behaviors and decision support must be adaptive to changes in the environment, and often behaviors and decisions must be coordinated with other units or agents.

We have developed a cognitive agent architecture that builds a framework for adaptive control and coordinated decision support. The Engine for Composable Logical Agents with Intuitive Reorganization (ECLAIR) incorporates the main mechanisms from Piaget's Cognitive-Stage Theory of Development [21], and it uses concepts from Damasio's Somatic Marker Hypothesis [6] to discover what should

* Work done while at ATL

be learned. 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), in which the comfortable level is a range, not a threshold. Agent wellbeing is an emotional state that is computed as a function of the agent's homeostatic vectors.

In traditional logistic systems, plans for action were predefined and static [24]. In normal operation, a pre-set plan may be suitable as it gives agents a guide for consistent behavior. However, in SRL, "normal" operation is often interrupted by events such as the appearance of a new adversary. In these cases, the need for dynamic re-planning is clear, and is provided in our agent architecture. Yet sometimes, even generating a new plan can be too time consuming for immediate survival. In these cases, our agents use adaptive, reflexive behavior that allows them to respond faster to unexpected or drastic changes in the environment, such as a loss of a supply unit when an engaged unit is dangerously low on ammunition. If an agent's perception of wellbeing indicates an urgent situation, reflexes will be fired in order to elicit immediate attention.

Adaptive learning extends both cognitive and reflexive behavior in our architecture. Cognitive adaptivity involves learning parameter and structure modifications for improved agent workflows using a genetic programming approach. Reflexive behaviors are adjusted to adapt to dynamic changes in the environment using a technique based on reinforcement learning [15, 27, 26]. Our reward function compares expectation of the reflex behavior versus actual observations, including change in the agent's emotional state. ECLAIR was developed as an extension to the Extensible Mobile Agent Architecture (EMAA) that has been applied to many military and DARPA applications [2, 3]. We demonstrated the cognitive architecture and reflexive adaptation using a simulation of net-centric warfare logistics and showed that agents are able to adapt their reflexive behavior to compensate for unexpected events in the environment.

This paper is organized as follows. Section 2 discusses related work in agents in military applications and logistics. Section 3 introduces Sense and Respond Logistics and describes its background, requirements, and challenges. Section 4 then discusses why agents represent SRL well, and what is required by the agent system to be effective for SRL. Section 5 describes our agent architecture and details our approach to plan and reflex adaptivity. Section 6 describes our logistics application and shows the results that indicate that agent adaptivity improves speed of command. We conclude with future work in Section 7 and concluding remarks in Section 8.

2. Related Work

2.1. Agents in Military and Logistics Applications

Adaptive agents is a well-studied topic that spans many approaches and domains [18, 7, 10]. Other agent systems exist that simulate the military domain and deal with the problems in it, but none of these systems approach the domain with the modern view of Net-Centric Warfare and Sense and Respond Logistics. TacAir-SOAR [14] is an expert system-based agent application for automated flight control and battlefield simulation developed using the rule-based, cognitive system SOAR. This system may be well-suited to the previous military application models that had completely pre-defined knowledge and problem models, but would not adapt well to 4GW. In today's battlefield environments, the environment and the adversarial agents in it cannot be completely modeled and any existing rules must be adaptive to environment changes. Unfortunately, sophisticated as it is, TacAir-SOAR has become obsolete for modern battlefields because it is not flexible.

Another agent system for battlefield simulation is the University XXI project [12]. This system begins to tackle cooperation amongst units, but it deals with larger units at the battalion level, not small, mobile units. A transition in military thought is occurring that believes that the difficulty is in controlling lower level units, while control at a higher level (tactical strategy) is both more understood and more able to be controlled by human commanders [24]. This system also uses the rules pre-built into it for all behaviors. Although it is reactive, it is not adaptive.

The Advanced Logistics Program (ALP) was initiated in 1996 as a five year DARPA/DLA project [11]. Its beginnings were rooted in the exploration of logistics planning and execution during Operations Desert Shield and Desert Storm. It was theorized that if information systems had better been able to handle specific logistics problems such as scheduling and coordination, then significant improvements would have been possible in resource sequencing and overall control over the supply chain. Thus, the challenge for ALP was to develop the technology to support an end-to-end logistics system with automated plan generation, execution monitoring, end-to-end movement control, and rapid supply and sustainment.

To address this challenge the ALP team developed the ALP agent architecture. This architecture provided advanced research into the areas of cognitive agency (capturing models of human cognitive processes in agents), fine-grained information management (techniques designed to minimize information propagation), and component-based design. Core pieces of the ALP architecture were made publicly available as open source as the COGnitive Agent ARchitecture (COUGAAR) [11].

Lockheed Martin's Advanced Technology Laboratories (LM ATL) has developed agent technology that offers promising solutions to the problems underlined in SRL. ATL has applied agent technology in more than two dozen projects covering a full range of intelligent systems, including information management for time-sensitive strike, situation awareness for small military units, and execution of user requests entered via spoken language dialogue [4, 5, 8, 19, 9, 20]. ATL's agents were also used in Navy Fleet Battle Experiments (FBE) as human aiding tools [20].

2.2. Cognitive Architectures

The two leading cognitive architectures with a psychological basis are SOAR [17] and ACT-R [1]. Both are hypotheses for answering Allen Newell's concept of a Unified Theory of Cognition [23]. Newell saw that in a person, there are many interacting components that must be integrated into a single comprehesive 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 and SOAR were developed based on contemporary psychological experimental results, and were not built on previous developmental theories.

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 in order to 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 interaction between these modules happens by each module exposing part of its activity into a buffer. The central production system uses the data in the buffers for its processing. ACT-R has primarily been used for psychological research, but has also been used to simulate computer generated forces for training purposes [1].

SOAR is a cognitive architecture focused on the functional requirements of human level intelligence. The three constraints that SOAR attempts to satisfy are goal driven behavior, continuous learning from experience, and showing "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 [22]. SOAR's mechanism for learning, called "chunking," has proven to cause unexpected results, and in many systems, such as TacAir-SOAR, has been turned off. Recently, experiments have been done to add reinforcement learning techniques to SOAR in place of the "chunking" mechanism [22].

While developing the ECLAIR architecture, ACT-R and SOAR were considered as possible starting points, but both were found to have a different strategy from our approach. Table 1 in Section 5.1 shows a comparison of our approach with that of SOAR and ACT-R. ECLAIR is further discussed in Section 5.1.

3. Sense and Respond Logistics

Net-Centric Warfare aims to combine "information-age concepts in the evolving strategic environment, enabling dispersed, semi-autonomous combat capability packages that produce coherent, mass effects via speed and coordinated efforts." [24] Today, many such 'capability packages' exist as Net-Centric Applications (NCAs) that run on the Global Informational Grid (GIG) and process all types of data (eg. HUMINT, SIGINT, IMINT, MASINT, OSINT, and GEOINT). When NCAs are integrated, they achieve novel capabilities in information delivery throughout a coalition environment.

Sense and Respond Logistics (SRL), is the process that handles the supply chain in 4GW. In NCW, sustaining operating tempo (OPTEMPO) is as much a logistics issue as it is kinetic. In order to maintain appropriate warfighting capability levels, the supply chain must not be interrupted. Unplanned operational pauses due to logistics problems are considered planning and adaptability failures. As the battlespace becomes ever more complex, the need for agile, robust logistics support of warfighter maneuvering becomes more crucial. As a result, current logistics planning is quickly becoming obsolete. Increasing numbers of asynchronous threats and specialized missions have caused the logistics problem to evolve. New problems are occurring including high rates of operational change, closely coupled events, unit-to-unit de-synchronization, and unacceptable speed of command. SRL must operate in an uncertain environment in which actions that have a positive effect today may not have the same results tomorrow.

For these reasons, Sense and Respond Logistics calls for new technology requiring, "adaptive responsive demand and support networks that operate in alternate structures that recognize operational context and coordination." [24] The solution to these problems requires dynamic network flexibility using properly orchestrated NCAs in order to deliver the right goods at the right times in a highly unstable environment. SRL aims to improve the speed that goods and services reach operational units in the battle theater. SRL is meant to be inclusive of coalition partners and responsive to unpredicted events.

4. Agent Systems for SRL

According to the United States Department of Defense [24], a networked, heterogeneous multi-agent system is needed to support Sense and Respond Logistics. These agents should represent all roles in the logistics domain, including the operational units (consumers), suppliers, and assets. In our agent architecture, roles are developed by supplying default stimuli and motivation to initiate action, and plans and reflexes to handle action. We are able to completely separate the agent architecture from the domain-specific extension. Methodologies for defining the specifics of agent behavior are provided for the scenario developer.

Another SRL system requirement is automated aids to support cognitive decision making [24]. These aids can take the form of automated control by the agent or agent-assisted decision support for the warfighters. In this paper we discuss methods, based on a cognitive agent architecture, to provide automation for low-level control normally handled by humans. This frees warfighters to concentrate on more complex aspects of warfare. Our command and control is in the form of tasks for an agent. A task is a unit of action, for example an action to move or make a request. A plan or workflow is a series of tasks, and a reflex is a single task.

The other aspect of decision support involves supplying a user with information and options in the form of a recommender system. This will aid the user with decisions that still need to be made by a human. The recommender system can be an extension of the automated agent control system. An agent will use the same decision process to find the best plan of action, but instead of completing the task autonomously, it will supply weighted options to the user and complete the task based on the user's input. We will be focusing on this capability in our future work.

SRL is considered to be a complex adaptive system [24]. Agents can adapt by reorganizing to suit the environment, or by modifying their behavior to improve their effectiveness. Our agents currently modify behavior by adapting plans and reflexes to make the outcome of their actions more closely match the expected results. In this paper we discuss our approaches to adapting cognitive (plan-oriented) and reflexive behavior. Genetic programming (GP) [16] is used to adapt the parameters and structure of plan behavior. Genetic material is composed of a GP tree representing a plan or workflow. Fitness of the plan is collected while the plan is being executed and is used to modify or generate new plans that adapt to the environment.

Currently, agents react to their perception of internal state by firing reflexes. The goal of any agent is to keep its homeostatic vectors in a comfortable range. Any homeostatic state outside of this range will result in a task or series of tasks being performed. The expectation is that the task will be performed as indicated, for example a request for a resource results in receiving the resource. We also expect that the original problem that initiated the task will be ameliorated. If either of these expectations is not met, then the parameters for the tasks may be adjusted. We based our technique from reinforcement learning [15, 27], using reward from task results to determine the value of task parameters. We discuss our technique in further detail in Section 5.3.

5. Agent Architecture for Adaptivity

5.1. ECLAIR, A Cognitive Architecture

5.1.1. Motivation ECLAIR is a technology that supports speed and coordination between many dispersed forces. NCW at the sub-battalion level consists of a dynamic environment with rapidly changing missions and contingencies. An ideal situation for logistics is that everyone is supplied just in time. Hierarchical distribution systems have had little success when applied to a scenarios in which troops are supplied when they are expected to run out of supplies [24]. The solution to such a problem consists of bringing decision making for changing supply routes and determining priorities down to the squad and unit level. The optimization of logistics tasks in a sub-battalion NCW environment is an optimization problem with a moving target, a.k.a. a dynamic optimization problem. Learning has proven to be a good tool to deal with such a moving target using both reinforcement learning and genetic programming. In addition, ECLAIR has a reflex learning system which reacts quickly to sudden changes in the environment.

Basic machine learning techniques require both careful selection of the necessary features for learning as well as a function to determine whether learning is improving behavior [25]. In a supervised learning scenario, the selected features are the inputs to the learning algorithms, and the function is based on a difference between the right answer and the output that the learning system gives. Given the dynamic nature of the Net-centric environment, it is unlikely that a general fitness function could be created, and even more unlikely that the set of features useful for learning could be determined ahead of time. The ECLAIR architecture uses a cognitive model to help determine the fitness and relevant features while processing, using a measure of well-being.

5.1.2. Theoretical Background 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 in the architecture are Piaget's adaptation theory [21], and Damasio's Somatic Marker Hypothesis [6]. Piaget's adaptation theory consists of three main concepts: *Assimilation, Accomodation*, 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. Finally, *Equilibration*, balances the aforementioned processes. The *Somatic Marker Hypothesis* stems from Damasio's belief that reasoning is not the only basis for decisions, but instead they are made based on gut feelings. A somatic marker is defined by Damasio as a trigger that recalls feelings related to the available 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 which is used as a map from circumstance to action.

5.1.3. Perception Our cognitive model approaches the processing problem from the perspective of interaction with the environment. This is similar to the Observation, Orientation, Decision, and Action (OODA) loop model used in military operations [28]. The main difference is that our model has a clear representation for learning and development, while the OODA loop does not. We separate our model into two interdependent processes. One is the decision loop which closely reflects the OODA loop, and the second is the adaptivity loop which is observing the decision loop until needed (Figure 1).

Instead of Observation, a cognitive model has *perception*. Perception affects both external and internal features. For a living being, external perception is in the form of sensed sound, smell, sight, taste, and touch, while internal perception includes hunger, pain, and comfort. ECLAIR's Stimulus and Awareness Modules interact within an agent to create stimuli from external and internal perceptions.

Orientation for a cognitive model occurs through interaction between perception, attention, and memory. In ECLAIR, when the Awareness module receives a stimulus, it matches the stimulus with reflex and plan behaviors that have the stimulus type as a condition. In addition, the stimulus updates the self and world representations. Part of these representations are Homeostatic Vectors(HVs), representing ideal levels of operation for different aspects of the agent. HVs are a mechanism for having multiple goal states, where the goal is for all states to be at an ideal level. The further away each variable is from its ideal level, the lower the wellbeing of the agent is. Orientation links the model of the world with the available behaviors and creates the basis for decision making.

5.1.4. Action Damasio's Somatic Marker Hypothesis makes light of an ongoing debate about the nature of decision making. Based on this debate, decision making appears to lie on a spectrum from reasoning to reaction. ECLAIR combines the decision making behavior of two methods in order to decide on an action. 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 it is not inhibited. If a plan is enacted, the plan will continue unless a higher priority plan is started. In the OODA loop, the final step is the Action. For a human, examples of action are speaking, moving one's self, manipulating an object and glancing. Agents on the other hand may be sending data, retrieving data, computing, sending control commands, ordering supplies, et cetera. The action is encapsulated in an activity chosen during the decision making stage.

As stated previously, learning does not have a firm place in the OODA loop. Orientation would be the easiest place to add learning to the OODA loop because in this stage, one can observe of all that is happening. However, there are a couple of problems with this. The first is that learning would slow down the orientation process, and the second is that learning during the orientation step does not allow for learning in one of the other three steps. In addition to having the decision loop in ECLAIR, there is a separate learning process that happens in parallel. This process allows for learning to happen at any stage of the decision process, and allows the decision loop to process without an expensive learning step. Figure 1 illustrates the connection between the decision loop and the adaptivity loop.

5.1.5. Adaptivity ECLAIR's Goal Based Adaptivity module can assist learning at any stage. The module is dedicated to improving each stage of the OODA loop by modifying and extending functionality. The Adaptivity Module listens to events from all other modules that are relevant for learning. The agent adapts in the Observation stage as input comes in through the stimulus module. The Awareness Module matches what it can and puts the unmatched input in a queue for later processing. The queue of unmatched input is one starting point for *Assimilation*. The input that doesn't match can be converted into its closest match, and then processed as if it were a known input.

As performance decreases, the HVs will move away from their ideal levels, causing low wellbeing. This will trigger adaptivity to find a better action. Eventually, the agent will *Accommodate* to the previously unmatched input. During the *Assimilation* phase, the agent adapts its orientation. Rather than setting aside the input, the agent translates the input into a form that can be processed. During the Accommodation phase, the agent creates a modified action for the input. This requires small changes to; orientation, since the inputs have to be distinguished; decision, since a new input condition has to be matched to a behavior; and action, since a new action or set of actions may be required to successfully accommodate to the novel input. The successful accomodation will be recognized by the agent through its wellbeing improving.

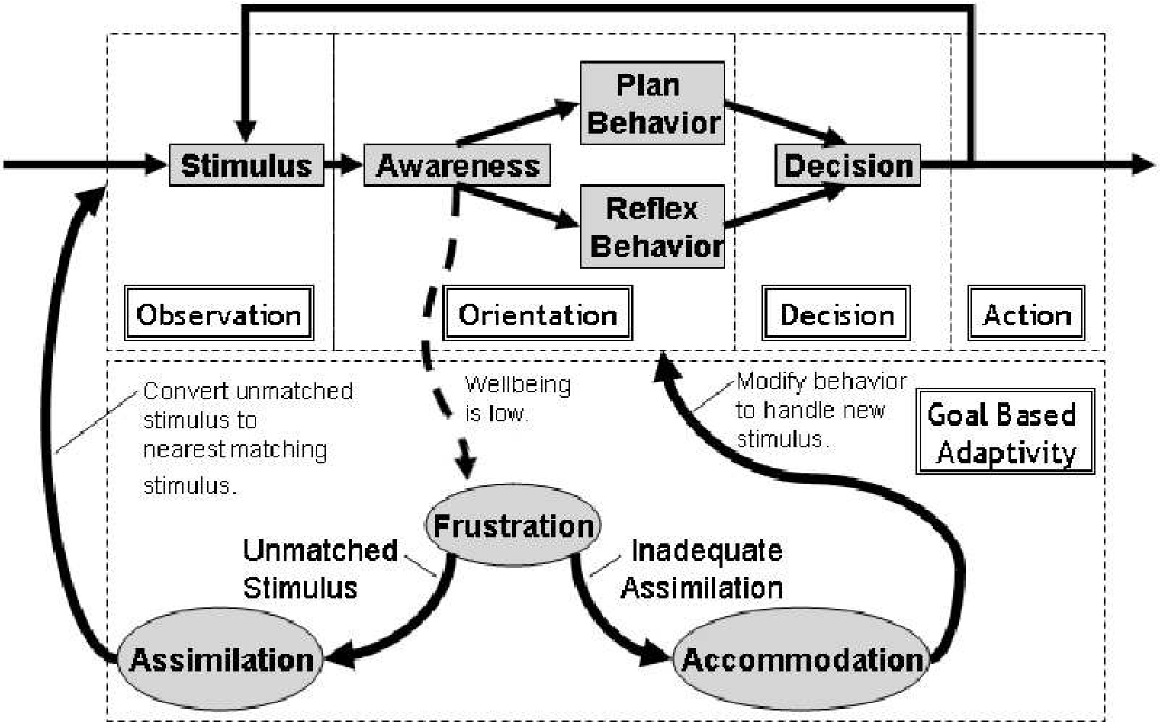


Figure 1. The ECLAIR Process Loop. The Goal Based Adaptivity loop listens to the decision loop during processing. When the agent wellbeing goes down, the agent becomes frustrated and determines the source of frustration. Once the source ist determined, either Assimilation or Accomodation processing is enacted to adapt to the frustration.

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 two such learning mechanisms. A genetic programming mechanism is discussed for plan adaptivity and a reinforcement learning based approach was implemented for reflex adaptivity.

Table 1 compares ECLAIR with ACT-R and SOAR. ECLAIR focuses on the control of the learning process while the other models attempt to model human behavior. ECLAIR also makes a distinction between behaviors that are part of a plan, and purely reflexive behaviors.

5.2. Plan Adaptivity

An ECLAIR agent's plan, also called a workflow, is a list of tasks linked by execution paths that can be conditional or unconditional. Tasks on an unconditional path are always executed, while tasks on a conditional path are executed only if the condition is met. Each task can take a certain number of task dependant parameters. The plan adaptivity mechanism was designed for this type of workflow.

Our approach to plan adaptivity is named Evolutionary Platform for Agent Learning (EPAL) and was described in detail in [3]. Genetic programming (GP) invented by John Koza [16] constitutes the basis for adaptivity in EPAL. GP uses the principles of Darwinian evolution for performing 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 how good a given individual is. 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. The method developed 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 as well.

	ECLAIR	SOAR	ACT-R
Maps to Brain function	No	No	Yes
Use to compare with human experiments	No	Yes	Yes
50 ms minimum process timing	No	Yes	Yes
Production System	No	Yes	Yes
Sub-symbolic level	Pending	No	Yes
Learning Methods	Genetic Programming, Reinforcement	Chunking of Productions or Reinforcement Learning	Utility Learning, Production Learning
Cause of learning	motivated by frustration	Chunking happens when declarative memory is specialized, RL happens from numeric preference rules	happens as part of storing memories.
Learning Switch	Wellbeing	Determined before run	Always on
Distinction between planned and reflexive action	Yes	No	No
Theoretical Basis	Cognitive Development	Functional Requirements	Integration of Components

Table 1. ECLAIR is compared with SOAR and ACT-R, the two leading cognitive architectures.

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 adaptivity in a logistics scenario yet, although we are currently integrating EPAL into ECLAIR's plan and adaptivity modules. We have demonstrated EPAL's operation and usefulness in a scenario similar to Fleet Battle Experiment-Juliet (FBE-J). Our experiment showed that agents learned to match sending rates of messages with the urgency of the messages to generate plans that improve overall network performance [3].

5.3. Reflex Adaptivity

A reflex in ECLAIR is composed of a stimulus, an activity, and a set of parameters for the activity (Figure 2). We use an approach based on reinforcement learning (RL) [15, 27] to learn the best parameters to use in an activity given the stimulus. Reinforcement learning 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 problem calculates the overall values of stimulus-activity-parameter mappings from the reward received as the results of activities are observed.

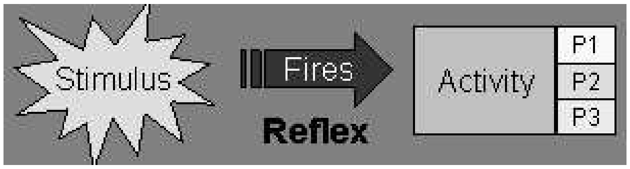


Figure 2. A reflex in ECLAIR. Contains a stimulus, activity, and parameters to the activity.

A typical reinforcement learning problem is composed of a set of discrete states, S, and a set of discrete actions, A. The high-level goal is to learn the best mapping between state and action ($s \to a$, $s \in S$, $a \in A$), or the best *policy*. In our architecture, a state is composed of a stimulus and an activity. Stimuli in our logistics scenarios include internal states (represented as HVs) such as LOW_FUEL and VERY_LOW_FOOD. Given these internal states, our agents will take an action; for example, ORDER_FUEL and ORDER_FOOD, respectively. We create S from combinations of stimuli and activities. Formally, $S \subseteq \Sigma \times \Lambda$, where Σ are all possible stimuli and Λ are all possible activities. In our current logistics application, S is pre-defined, however, *Accomodation* could be used to extend S.

The parameters to the activity, for example who to order from, how much to order, and what priority the order should be, are variable and constitute our learning problem. We create A from the occurring combinations of parameters: $A = \bigcup_{i=1}^{m} \Phi_i, \Phi_i = P_1 \times P_2 \cdots \times \cdots P_j \cdots \times \cdots P_n$, where P_j is a parameter type and all its values, n is the number of parameter types for activity i, and m is the number of activities. Our policies are then composed of $\{$stimlus-activity$\}$-$\{$parameter set$\}$ mappings, corresponding to RL's state-action ($s \rightarrow a$) mappings: $s = \{\sigma, \lambda\}$, $a = \phi$, where $\sigma \in \Sigma$, $\lambda \in \Lambda$, and $\phi \in \Phi_i$. Our adaptivity module for reflex behavior stores overall values for policies that it computes over time.

Reinforcement is computed by comparing the expectation of the activity with the observations that are seen as a result of the activity occuring. This was the main challenge in our approach as the observations from an activity are not immediate and may not be seen until several intermediate tasks are completed. For example, in our logistics simulation, the expectation from ORDER_FUEL is that we will receive the amount of FUEL we ordered *within a certain period of time*. In order for an agent to receive a resource, a supplying agent must receive the order for the resource, 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.

When an agent fires a reflex, its awareness module generates an *expectation* object that indicates the expected results, as well as a time that the result should be expected by. When the agent receives a stimulus, it generates an *observation* object if the stimulus is of a type that it is interested in. The agent then attempts to match the *observation* with an *expectation* using an ID that may indicate that the *observation* occurred as a result of the reflex being fired (the reflex that generated the expectation). For example, a RECEIVED_FUEL observation may have occurred because of an ORDER_FUEL reflex. If a match occurs between the *observation* and an *expectation*, the agent's adaptivity module then compares the details of the *expectation* with the *observation* to generate an *expectation* versus *observation* (XVO) value. Figure 3 shows the process flow for reflexes. The XVO value is between -1.0 (does not meet expectations) and 1.0 (meets expectations). If the *observation* does not occur within an extended period of time, an *expired observation* will be created, and XVO will be -1.0. If the *observation* occurred within the expected time, and had the correct parameters, then the XVO will be 1.0. Values between -1.0 and 1.0 could occur if the *observation* was late, or had only part of the requested resources. Just like motivation and behavior, the expectation object is configurable, so other methods for computing XVO could be used.

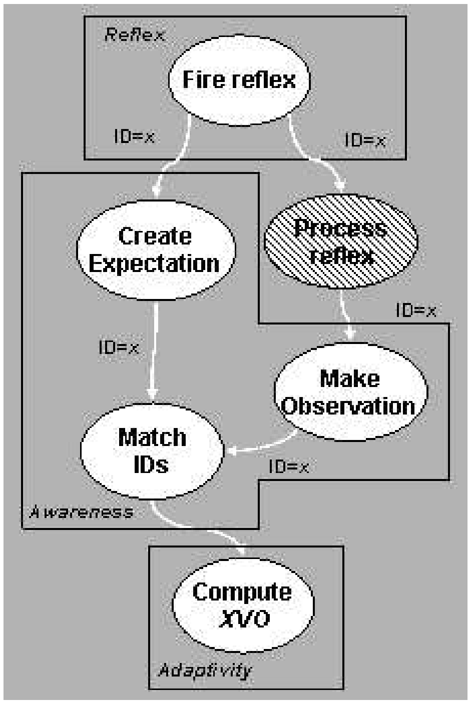


Figure 3. The process flow for a reflex in ECLAIR. The solid white nodes show the behavior of the agent firing the reflex. Boxes represent the modules that handle the nodes. The node with diagonal lines could be handled by other agents.

Reinforcement value is a function of the XVO and the change in homeostatic vectors that may occur due to a reflex being fired. This causes the awareness module to consider that even if the reflex yielded the expected results, it may not have been the correct approach if it did not improve our situation. Reinforcement is given to the policy that caused the *observation* to occur, as well as previous policies that have been attempted that used common parameters in the same situation. We are attempting to generalize reinforcement without over-fitting (reinforcing the wrong behavior). Formally, the reinforcement, r, for any policy that has been used while running the system is as follows:

$$r = \gamma^{-i}\alpha(XVO + \Delta HV){\frac{|P_p \bigcap P_o|}{|P_p|}}^{\sigma}$$

Where $i \equiv$ the age of the policy being rewarded (most recent=1), ΔHV = the change in homeostatic vector level (positive= improved, negative=deteriorated), P_p = the set of parameters in the policy's activity, P_o = the set of parameters in the observation, and γ, α and σ are learning-rate variables.

Overall value for a policy is the summation of its reinforcement, r, 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 wellbeing of the agent. If wellbeing is high, then we consider that the agent is doing well with his default or learned behaviors, and do not explore often. However, if wellbeing is low, then the agent explores more often in order to find better policies to improve wellbeing. Exploration is a key component of reinforcement learning as it allows the agent interacting with the system to try actions that it may not have tried if it was only considering the current "best" action [27]. It may be the case that a new action is better for a given situation than anything the agent has tried before. Balancing exploration with policy is also a difficult problem in RL. Our solution elegantly incorporates knowledge about our internal emotional state to compute an exploration rate that is well suited to the cognitive architecture.

6. Application

6.1. Logistics Simulation

ATL created a demonstration application that shows how ECLAIR agent adaptivity applies to logistics. The prototype shows solutions to two important SRL goals: ECLAIR agents improve the speed of command in a robust fashion and adapt to the changes in a demand driven network.

Figure 4 depicts the application's interface. ECLAIR agents represent three operational units (OU) (boxes with an X) and two supply units (SU) (boxes with a horizontal line). As an OU, the ECLAIR agent monitors its homeostatic states that indicate how much fuel, ammunition, and food it has. As the OU consumes its resources, it becomes increasingly unhappy until it is stimulated to request a re-supply. Re-supply requests are drawn as arrow-headed lines pointing to the SU the request was sent to. OUs set their expectations based on which supply unit they sent the request to, how much they requested, and how long they expect to wait for the request to be fulfilled. When supplied, the OU agent makes complementary observations on which supply unit delivered the resource, 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.

ECLAIR agents also represent supply units. Behavior for re-supplying supply units is similar to operational units, except that SUs will send an asset to a ship (circle labeled "AR") instead of sending a request to another unit. The assets include trucks (circle containing a box) and helicopters (hemisphere containing a bow-tie). For these simulations, we concentrated on adaptivity of OUs, although the capabilities to adapt SU behavior were available.

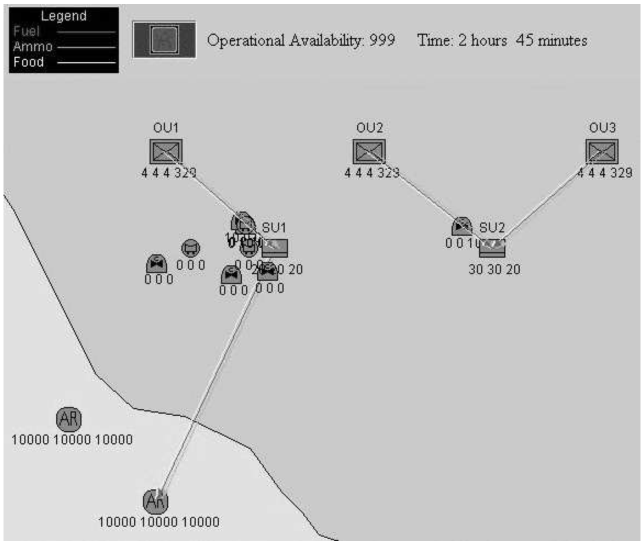


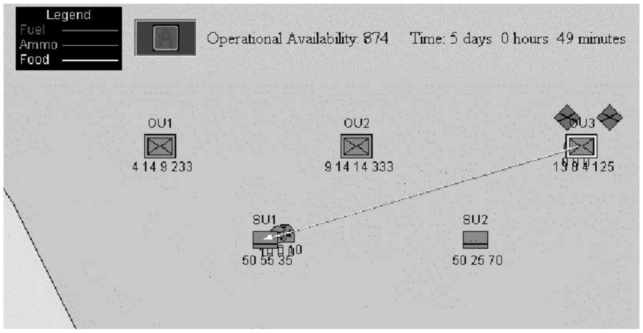
Figure 4. The ECLAIR logistics demonstration before adaptation (beginning of simulation). OU2 and OU3 order resources from default supply unit (SU2).

Figure 4 depicts a scenario where the left-most SU, *SU1* has nine assets and the right-most SU, *SU2* has only one. The OUs, (*OU1, OU2,* and *OU3*) are initially assigned a default supply unit to order resources from. By default, OU1 requests resources from SU1 and OU2 and OU3 request resources from SU2. In Figure 4, the arrows from OU2 and OU3 show requests for resources being made to SU2.

The demonstration uses the concept of Operational Availability (AO) as a metric in determining whether adaptation is truly occuring. AO measures how long every OU has to wait to be supplied. AO is determined by summing up the personal AO scores (PAO) of each operational unit. Each OU has a maximum PAO of 333, making AO a maximum of 1000 (rounded). The rate of PAO degredation depends upon what was requested (fuel ≫ ammunition ≫ food) and what state the OU was in when it made the request (engaged ≫ moving ≫ idle). In Figure 4, the PAO for OUs is the right-most number under the OU icons (boxes with an X). The first three numbers are the levels of fuel, ammunition and food.

6.2. Results

Figure 5 shows the typical results of the demonstration scenario. Since SU2 was "handicapped" in that it only had one asset, compared to SU1's nine assets, we expected that adaptation would cause all OUs to request most of their resources from OU1. In order to add an element of instability in the environment, enemy units (diamond with an X) periodically attacked OU3. Initially, AO decreased rapidly until the ECLAIR agents learned to choose different supply units based on the availability of resources. Within a short period,



Operational Availability: 874 Time: 5 days 0 hours 49 minutes

Figure 5. The ECLAIR logistics demonstration after adaptation. OU3 learns to order resource from SU1 because it is more reliable.

OU2 and OU3 learned to decrease the expectations of SU2's reliability because it had only one asset. Also, as wellbeing decreased, more exploration occurred, causing the OUs to send their requests to SU1. Eventually, OUs almost always requested from the SUs that gave the right types of resources in the shortest period of time. For example, Figure 5 shows OU3 requesting supplies from SU1.

In our demonstration prototype, we show that the ECLAIR agent framework provides solutions to SRL challenge problems; agents improve the speed of command and adapt to the changes in a demand driven network. Speed of command is measured by the AO score. Figure 6 shows the average AO score of 30 runs for three scenarios based on the nine-one asset scenario described previously; *Default*, *Explore*, and *Adapt*. The dark gray line, marked "Default," shows the results of agents only requesting supplies from their default supply units. The black line, marked "Explore," shows the results of agents selecting a random supply unit at an exploration rate based on wellbeing, but not using learning results to adapt. The white line, marked "Adapt," shows the results of agents using learning results to adapt. In the *Default* scenario, agents only used their default behavior. In *Explore*, the agents may have randomly chosen the best supply unit to request to, but they were not making selections based on learned knowledge. The AO scores for the *Adapt* scenario were considerably higher than the other scenarios. *Default* quickly bottoms out at the lowest possible score. *Exploration* only reaches an AO score of around 400, while *Adapt* flattens out near 1000, the maximum score.

The graph in Figure 6 shows that using learning to adapt to the environment yields a clear improvement in speed of command. 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 a more responsive SU1 for supplies. The speed of command was improved from its initial setting when SU2 was supplying all of the needs to both OU2 and OU3.

Our final scenario shows agent adaptation to dynamic changes in the environment. We set up a contrived scenario in which resource availability for supply units changed drastically over time. In the scenario, OU3 sent requests to SU1 by default, however SU1 initially had no ammunition and took several days to order more. SU2 had a stockpile of ammunition, but it could not send assets to get more when it ran out. In order to show that OU3 was adapting to the changes in the environment, it should have first learned to request resources from SU2, but should have later switched to SU1 after SU2 ran out of ammunition. Figure 7 indicates when (x-axis) and to whom (y-axis) OU3 sent requests for ammunition. The vertical lines indicate events that changed the environment in the scenario. The events are as follows:

1. Begin. SU1 has no ammunition, but SU2 has 75 units of ammunition
2. Enemy appears
3. SU1 receives 100 units of ammunition
4. SU2 runs out of ammunition and does not order more

OU3 did learn to adapt to the changes in the environment. At first it made four requests for ammunition from its default supplier, SU1, but then learned to make requests to SU2. After SU2 ran out of ammunition, OU3 explored and made a request to SU1 around day six. At around day nine OU3 learned to continue making requests to SU1, and did so almost exclusively (except for the exploration around day 14). This scenario shows that adaptivity occurs quickly enough to respond to frequent changes in the environment. For example, OU3 initially learned to request from SU2 after only four interactions with the environment, and then re-learned to order from SU1 after another four examples. Our goal for future work is to use learned knowledge from other agents and the environment to speed command even more.

7. Future Work

Our future work will extend our architecture by using agent cooperation to share learned behaviors and awareness of the environment. We will develop decision assistant agents that make weighted recommendations to small-unit commanders based on learned knowledge. We plan to investigate two agent research areas; coordination and sharing, in order to make agents more synchronized in their behaviors and knowledge of the world. Not only will agents adapt by modifying their individual behaviors, they will also re-organize themselves or their assets to better suit the environement.

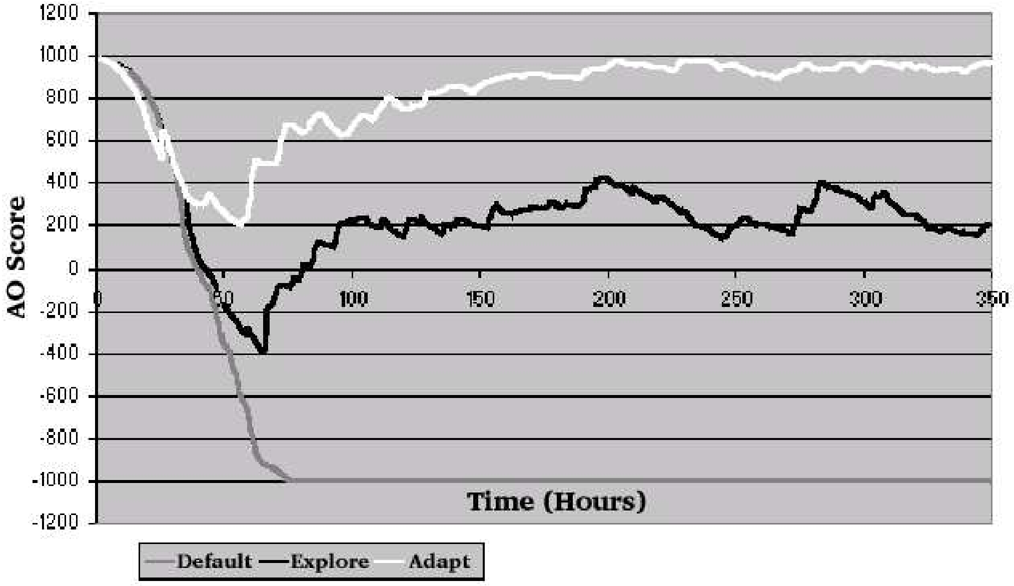


Figure 6. Average AO scores for 30 runs, over a period of 350 simulated hours. The dark gray line shows the score of the default behavior. The black line shows the score using some random exploration. The white line shows the score using adaptivity.

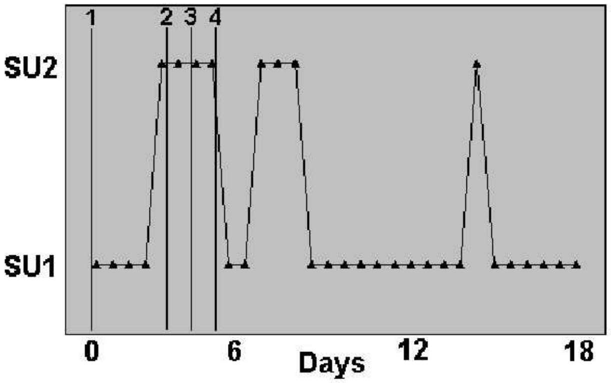


Figure 7. Results from a scenario depicting several events that cause OU3 to adapt its behavior. OU3 learns to send request to SU1 or SU2, depending on their resource availability

Anticipation is also important for SRL applications [24]. We will investigate the potential for agents to anticipate the needs of other agents. For example, detecting changes in consumption rates of operational units due to engagement with a new force should prompt a supply unit to prepare for delivery sooner than planned, or even cause it to recommend a new configuration of suppliers to better support engaged units. Currently the need for resources is determined by operational unit agents. In the future more responsibility will be moved to supplier unit agents. We will also study the interplay between plan and reflexive behavior, a subject that is not well-studied in agent research.

8. Conclusion

We have developed a cognitive agent architecture that provides the framework for Sense and Respond Logistics. SRL requires coordination amongst agile, responsive units, and presents an optimization problem with a moving target. Our agent architecture advances the current state of logistics applications because agents can follow the moving target by adapting to a changing environment. We presented a logistics implementation of our architecture that shows that adaptive agents have greatly improved behavior over agents that do not adapt.

Our agent architecture uses cognitive models based on Piaget's Cognitive-Stage Theory of Development [21] and Damasio's Somatic Marker Hypothesis [21]. Agents take a hybrid approach to action, using logic-based plan 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 increasing the speed of command. Our cognitive architecture is an excellent framework for SRL because plan behavior encourages agents representing warfighters to follow strategies built from experience in the battlefield, while reflex behavior helps the agents handle unexpected situations.

ECLAIR's unique contributions to agent research are the cognitively-inspired architecture that supports decision making using plan and reflexive behavior, and our net-centric approach to logistics. We are using adaptive agents to tackle the critical problems summarized by the Department of Defense [24] with an approach that is oriented towards SRL. ECLAIR bridges research in adaptive agents and cognitive architectures with the military domain that is just beginning to acknowledge the need for adaptive systems. We are also interested in making usable tools for real-life problems. At ATL, we have the proven capabilities to extend our research into the real world and supply practical applications for use by warfighters in the global theater.

References

[1] ANDERSON, J. R., BOTHELL, D., BYRNE, M. D., DOUGLASS, S., LEBIERE, C., AND YULIN. An integrated theory of mind. *Psychological Review 111*, 4 (2004), 1036–1060.

[2] BUCZAK, A. L., COOPER, D. G., AND HOFMANN, M. O. Evolutionary platform for agent learning. In *Proceedings of the Intelligent Engineering Systems Through Artificial Neural Networks* (New York, 2003), vol. 13, ASME Press, pp. 201–206.

[3] BUCZAK, A. L., COOPER, D. G., AND HOFMANN, M. O. Evolutionary platform for agent learning. In *Proceedings of the Intelligent Engineering Systems Through Artificial Neural Networks* (New York, 2004), vol. 14, ASME Press, pp. 157–164.

[4] COOPER, D. G. Context based shared understanding for situation awareness. In *Proceedings of the MSS National Symposium on Sensor and Data Fusion, 2004* (2004).

[5] CZAJKOWSKI, M., BUCZAK, A. L., AND HOFMANN, M. O. Dynamic agent composition from semantic web services. In *Proceedings of the 2nd Workshop on Semantic Web and Databases (SWDB), 2004* (2004), pp. 1–14.

[6] DAMASIO, A. R. *Descartes' Error: Emotion, Reason, and the Human Brain*. G.P. Putnam, New York, 1994.

[7] DECKER, K. S., AND SYCARA, K. Intelligent adaptive information agents. In *Working Notes of the AAAI-96 Workshop on Intelligent Adaptive Agents* (Portland, OR, 1996), I. Imam, Ed.

[8] FRANKE, J., SATTERFIELD, B., AND JAMESON, S. Information sharing in teams of self-aware entities. In *Proceedings of the The Second International Workshop on Multi-Robot Systems NRL* (2003).

[9] GERKEN, P., JAMESON, S., SIDHARTA, B., AND BARTON, J. Improving army aviation situational awareness with agent-based data discovery. In *Proceedings of the American Helicopter Society Conference* (2003).

[10] HAYNES, T., WAINWRIGHT, R., AND SEN, S. Evolving cooperation strategies. In *Proceedings of the First International Conference on Multi–Agent Systems* (San Francisco, CA, 1995), V. Lesser, Ed., MIT Press.

[11] HELSINGER, A., THOME, M., AND WRIGHT, T. Cougaar: a scalable, distributed multi-agent architecture. In *Systems, Man and Cybernetics* (2004), vol. 2, IEEE, pp. 1910–1917.

[12] IOERGER, T. R., VOLZ, R. A., AND YEN, J. Modeling cooperative, reactive behaviors on the battlefield using intelligent agents. In *Proceedings of the The Ninth Conference on Computer Generated Forces (9th CGF)* (2000), pp. 13–23.

[13] JOHNSON, M. H. *Developmental Cognitive Neuroscience: An Introduction*. Blackwell Publishers, Inc., Cambridge, MA, 1997.

[14] JONES, R. M., LAIRD, J. E., NIELSEN, P. E., COULTER, K. J., KENNY, P., AND KOSS, F. V. Automated intelligent pilots for combat flight simulation. *AI Magazine 20*, 1 (1999), 27–41.

[15] KAELBLING, L. P., LITTMAN, M. L., AND MOORE, A. W. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research 4* (1996), 237–285.

[16] KOZA, J. R. *Genetic Programming: On the Programming of Computers by Natural Selection*. MIT Press, Cambridge, MA, 1992.

[17] LEWIS, R. L. Coginitive theory, soar. Tech. rep., Ohio State University, Doepartment of Computer Science, 1999.

[18] LITTMAN, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)* (New Brunswick, NJ, 1994), Morgan Kaufmann, pp. 157–163.

[19] LOCKHEED MARTIN ADVANCED TECHNOLOGY LABORATORIES. *Cooperative Agents for Specific Tasks (CAST)*. http://www.atl.lmco.com/overview/programs/IS/CAST.html.

[20] LOCKHEED MARTIN ADVANCED TECHNOLOGY LABORATORIES. *Published Papers*. http://www.atl.lmco.com/overview/library.html.

[21] MILLER, P. H. *Theories of Development Psychology*. W.H. Freeman and Co., 1983.

[22] NASON, S., AND LAIRD, J. E. Soar-rl: Integrating reinforcement learning with soar. Tech. rep., University of Michigan, 2004.

[23] NEWELL, A. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, 1990.

[24] OFFICE OF FORCE TRANSFORMATION, UNITED STATES DEPARTMENT OF DEFENSE. *Operational Sense and Respond Logistics: Coevolution of an Adaptive Enterprise Capability*, 2004. Concept document in progress.

[25] RUSSELL, S. J., AND NORVIG, P. *Artificial Intelligence: a modern approach*, 2nd international edition ed. Prentice Hall, Upper Saddle River, N.J., 2003, ch. 18: Learning from Observations, pp. 649–651.

[26] SUTTON, R. S. Reinforcement learning: Past, present and future. In *SEAL 1998* (1998), pp. 195–197.

[27] SUTTON, R. S., AND BARTO, A. G. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[28] WOOD, R. J. Information engineering; the foundation of information warfare. Tech. rep., Air War College, Air University, 1995.