Cognitively inspired anticipatory adaptation and associated learning mechanisms for autonomous agents

A. Negatu

S. Mello

S. Franklin

Cognitively Inspired Anticipation and Anticipatory Learning Mechanisms for Autonomous Agents

Aregahegn Negatu, Sidney D'Mello, and Stan Franklin

Department of Computer Science and the Institute for Intelligent Systems
The University of Memphis
Memphis, TN 38152, USA
{asnegatu|sdmello|franklin}@memphis.edu

Abstract. This paper describes the integration of several cognitively inspired anticipation and anticipatory learning mechanisms in an autonomous agent architecture, the Learning Intelligent Distribution Agent (LIDA) system. We provide computational mechanisms for variants of payoff, state, and sensorial anticipatory mechanisms. The payoff anticipatory mechanism in LIDA is implicitly realized by the action selection dynamics of LIDA's decision making component. A description of a non-routine problem solving algorithm is presented as a form of state anticipatory mechanism. A technique for action driven sensational and attentional biasing is offered as a viable sensorial anticipatory mechanism. We also present an automatization mechanism and an instructionalist based procedural learning algorithm as forms of implicit and explicit anticipatory learning mechanisms.

1 Introduction

While the role of anticipations on deliberation, memory, attention, behavior, and other facets of cognition has been well studied in cognitive psychology, neuropsychology, and ethology, the literature on explicit mechanisms to realize anticipations in artificial agents is considerably more sparse and scattered (Blank, Lewis, & Marshall, 2005; Butz, Sigaud, & Gerard, 2002; Kunde, 2001; Rosen, 1985; Schubotz & von Cramon, 2001). Since anticipations have been acknowledged to be an influential component of the cognitive facilities of humans (and other animals), the need to model and integrate theories of anticipations in our artificial systems becomes vital. Over the last decade a variety of mechanisms that realize anticipations in artificial systems have been proposed (e.g., Blank, Lewis, & Marshall, 2005; Drescher, 1991; Stolzmann, 1998; Witkowski, 1997). Butz, Sigaud, and Gerard (2002) provide several examples of such systems and have devised a useful nomenclature for the various anticipatory mechanisms that include *payoff*, *sensorial*, *state, and implicitly* anticipatory systems. The fundamental difference between implicit and the other three anticipatory systems is that in implicitly anticipatorial systems no explicit predictions about the future are made, even though the structure of the action selection component must contain certain anticipatory elements. Sensorial anticipation differs from payoff and state anticipatory mechanisms in that the predictions influence both early and later stages of sensory processing without directly having a direct impact on action selection. Finally, the main difference between payoff and state anticipatory mechanisms is that in payoff anticipatory systems anticipations play a role as payoff predictions only and explicit predictions of future states are not made. On the other hand state anticipatorial mechanisms make explicit predictions of future states during decision making processes.

Our interest in anticipation and anticipatory learning mechanisms emerges from a desire to model several facets of human (and animal) cognition in an autonomous agent the Learning Intelligent Distribution Agent (LIDA). LIDA is the partially conceptual, learning extension, of the original IDA system implemented computationally as a software agent (D'Mello et al., 2006). The original IDA system was designed as an autonomous agent and performed personnel work for the US Navy in a human-like fashion (Franklin, 2001). Although the design of IDA was inspired by several theories of human and animal cognition, it did not learn. The LIDA system adds three fundamental forms of learning to IDA: perceptual, procedural, and episodic learning.

In this paper we describe computational mechanisms for several types of anticipatory mechanisms, as used in the LIDA model. Sensory anticipation is accomplished in LIDA via a preafferent signal (Freeman, 1999), sent upon the decision to take an action, that biases LIDA's perceptual mechanism in favor of the anticipated sensory information (see 3.3 below). Payoff anticipation is implemented implicitly by LIDA's action selection mechanism, as the next action to be taken is chosen (see 3.1). The non-routine problem solving algorithm (see 3.2 below) produces state anticipation in LIDA. Mechanisms for anticipatory learning are also described (see 4. below).

The anticipatory mechanisms described for the LIDA model may well be suitable for incorporation into other cognitive based computational architectures designed to control autonomous software agents and/or mobile robots. If experience with humans and animals is any indication, such anticipatory mechanisms are likely to prove of substantial value to software agents or mobile robots in complex, dynamic environments. Also, its inclusion of several anticipatory mechanisms adds to LIDA's value as an animat system.

We begin by briefly describing components of the LIDA architecture that play a central role in realizing payoff, state, and sensorial anticipatory mechanisms. We then describe the underlying processes that implement these three anticipatory mechanisms in LIDA. Next we outline computational mechanisms for implicit and explicit anticipatory learning, i.e., the learning of what to anticipate.

2 Architectural Support for Anticipation

The LIDA architecture is partly symbolic and partly connectionist with all symbols being grounded in the physical world in the sense of Brooks (1986). The fundamental computational mechanism of the LIDA system is the *codelet* (Hofstadter & Mitchell, 1994), a small piece of code executing as an independent thread that is specialized for some relatively simple task. The components of the LIDA system that are related to anticipations and anticipatory learning include perceptual associative memory, selective attention, procedural memory, and action selection. Other components, being only peripheral to this paper, are not described here.

2.1 Perceptual Associative Memory

The perceptual knowledge-base takes the form of a semantic net with activation (called the slipnet) motivated by Hofstadter and Mitchell's Copycat architecture (1994). Nodes of the slipnet constitute the agent's perceptual symbols (Barsalou, 1999), representing individuals, categories and simple relations. The perceptual symbols are grounded in the real world by their ultimate connections to various primitive feature detectors having their receptive fields among the sensory receptors. An incoming stimulus, say a visual image, is descended upon by a hoard of perceptual codelets. Perceptual codelets respond to specific features from the various sensory streams and perform perceptual tasks such as recognition and identification. Each of these codelets is looking for some particular feature (a certain color, a line at a particular angle, etc) or more complex features (a T junction, a red line). Upon finding a feature of interest to it, the codelet will activate an appropriate node or nodes in the slipnet. Activation is passed. The network will eventually stabilize. Nodes with activations over threshold, along with their links, are taken to provide the constructed meaning of the stimulus, the percept (see Figure 1).

2.2 Selective Attention

Selective attention in LIDA is an implementations of Global Workspace Theory (Baars, 1988) with hosts of attention codelets that each play the role of a daemon watching for an appropriate condition under which to act. Attention codelets form coalitions (collection of related codelets) with other codelets to compete for attention. Upon noting a suitable situation, an attention codelet will increase its activation as a function of the match between the current situation and its preferences. During any given cycle one of these coalitions with the highest average activation is considered relevant and broadcasts its information to every other codelet (Baars, 1988). This broadcast is used to recruit schemes (see below) and perform various types of learning (D'Mello, et al., 2006).

2.3 Procedural Memory

Procedural memory in LIDA is a modified and simplified form of Drescher's schema mechanism (1991), the scheme net. The scheme net is a directed graph whose nodes are (action) schemes and whose links represent the 'derived from' relation. Built-in primitive (empty) schemes directly controlling effectors are analogous to motor cell assemblies controlling muscle groups in humans. A scheme consists of an action, together with its context and its result (see Figure 1). The context and results of the schemes are represented by perceptual symbols (Barsalou, 1999) for objects, categories, and relations in perceptual associative memory. The action of a scheme consists of one or more behavior codelets (discussed next) that execute the actions in parallel.

2.4 Action Selection

The LIDA architecture employs an enhancement of Maes' behavior net (1989) for high-level action selection in the service of drives. The behavior net is a digraph (directed graph) composed of behavior codelets (a single action), behaviors (multiple behavior codelets operating in parallel), and behavior streams (multiple behaviors operating in some partial order) and their various links. These three entities all share the same representation in procedural memory (i.e., a scheme). As in connectionist models, this digraph spreads activation. The activation comes from three sources: from pre-existing activation stored in the behaviors, from the environment, and from drives. To be acted upon, a behavior must be executable (preconditions satisfied), must have activation over threshold, and must have the highest such activation.

LIDA's action selection mechanism incorporates five major enhancements over Maes' behavior net: (i) *Variables* – While Maes' behavior net operates on the basis of boolean propositions only, LIDA's mechanism supports variables that get bound during the instantiation of procedural schemes; (ii) *Restricted search space* – During the action selection phase Maes' mechanism performs a global search over all the available competency modules while the enhanced behavior net restricts its search to relevant (instantiated) goal hierarchies, which are a subset of the available competencies; (iii) *Failure handling* - Maes' mechanism assumes that the result of a selected action is deterministic in that every action produces its expected outcome. Therefore, this mechanism is unable to handle execution failures which frequently occur in any real system. On the other hand LIDA's enhanced behavior net is endowed with a degree of fault tolerance via its expectation mechanism; (iv) *Priority control* – Maes' mechanism modulates the priorities of competing goals by building *static* causal links among competence modules while LIDA's mechanism provides parametric control to *dynamically* change goal priorities at run time; (v) *Planning and subgoaling* – Maes' mechanism does not support classic AI planning and subgoaling but LIDA's mechanism, as a collection of goal structures, supports both (see Negatu, 2006).

2.5 LIDA's Cognitive Cycle

Since the LIDA architecture is composed of several specialized mechanisms the need for a continual process that causes the functional interaction among the various components becomes paramount. We offer the cognitive cycle as such an iterative, cyclical, continually active process that brings about the interplay among the various components of the architecture. A complete description of the cognitive cycle can be found in Franklin et al. (2005). We restrict our discussion to the four major components described above as follows.

The meaning of an incoming stimulus is constructed in perceptual associative memory and is taken to be nodes that are above a certain threshold. The attention codelets build coalitions among these nodes and compete for attention. The contents of a winning coalition are broadcast to procedural memory to instantiate action schemes. Instantiated schemes compete for execution in the behavior net as behaviors. The dynamics of the behavior net select an action and the agent then directs its focus to perception.

3 Anticipatory Mechanisms

The LIDA architecture includes payoff, state, and sensorial anticipatory mechanisms as outlined by Butz, Sigaud, and Gerard (2002). At this stage, the payoff and the preparatory attention component (described below) of the sensorial mechanisms have been computationally implemented. We now briefly describe these mechanisms. A graphical depiction of the sensorial and payoff anticipatory mechanisms embedded with LIDA's cognitive cycle is presented as Figure 1.

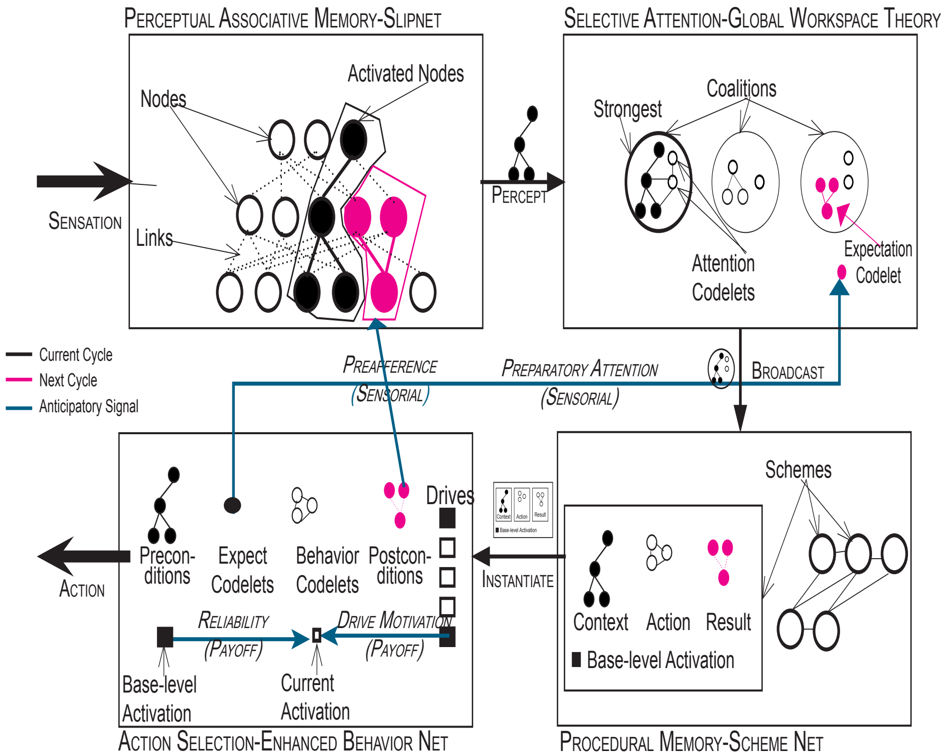

Figure 1. LIDA's Cognitive Cycle and Payoff and State Anticipatory Mechanisms

3.1 Payoff Anticipatory Mechanisms

In a payoff anticipatory mechanism no explicit predictions of future states are made with the role of anticipations being restricted to some form of payoff, or utility, or reinforcement signal. In the LIDA model the payoff for a behavior is calculated on the basis of predictive assessments by its *current activation* (i.e., relevance to the current goals or drives and environmental conditions) and its *base-level* activation (i.e., reliability in past situations).

LIDA's motivational system to influence goal-directed decision making is implemented on the basis of *drives*. Drives are built-in or evolved (in humans or animals) primary and internal motivators. All actions are chosen in order to satisfy one or more drives, and a drive may be satisfied by different goal structures. A drive has an importance parameter (real value in [0,1]) that denotes its relative significance or priority compared to the other drives. Each drive has a preconditional proposition that represents a global goal. A drive spreads goal-directing motivational energy, which is weighted by the importance value, to behaviors that directly satisfy its global or deep goal. Such behaviors in turn spread activation backward to predecessor behaviors.

Although external activation spreading includes situational motivation, in this discussion of anticipation, we will attend only to the action selection dynamics that are tuned to goal-end motivation. From this point of view, the current activation of a behavior at a given time represents the motivation level for its execution to satisfy sub-goals, which in turn contributes towards satisfying one or more global goals at some future time. In other words anticipating the predictive payoff in satisfying a goal influences the selection of the current action.

This payoff anticipatory mechanism was tested as a controller of Khepera robot in a simulated environment (a warehouse) with tasks of differing priorities (drive levels). As described above in order to maintain priorities for the competing tasks (goals), behavior streams (goal structures or partial plans of action) that satisfy the various goals have executable behaviors with motivation levels that reflect the priorities of the goals. In general, maintenance of priorities is possible if and only if, when considering all behavior streams with specified priorities (obtained from the importance parameter of the motivating drive), the selected behavior should belong to the one with the highest priority. All other things being equal between two behavior streams, the one motivated by a drive with a high importance parameter is expected to have a higher average activation level than the other motivated by a drive with a lower importance parameter. Figures 2 shows how goal-driven motivation levels serving as payoff anticipations vary with time. The test data was obtained by taking a snapshot of the activation/motivation levels of two behaviors that were parts of two competing instantiated behavior streams (a behavior stream is instantiated by the selective attention mechanism as described above). Figure 2a shows that in the absence of the importance parameter for the drives (as in the case of Maes' original mechanism), the payoff anticipatory mechanism could not be effectively tuned to control the priorities of the competing tasks as evidenced by the interleaving of the activation levels of the two behaviors. However, as illustrated in Figure 2b, LIDA's use of the importance parameter (for the drives) consistently increases the activation or payoff of the behavior with the higher priority. Additionally, the payoff anticipatory mechanism and the action selection mechanism of the enhanced behavior net in general were found to correlate highly with a simulated human operator (Negatu, 2006) using GOMS task analysis (Card, Moran, & Newell, 1983).

The second factor that influences the payoff in selecting an action involves the use of the base-level activation of a scheme, which is a uninstantiated behavior in procedural memory. The base-level activation is a measure of the scheme's overall reliability in the past, and is computed on the basis of the procedural learning mechanism described in the next section. It estimates the likelihood of the result of the scheme occurring after taking the action in its given context. When a scheme is deemed somewhat relevant to the current situation as a result of the attention mechanism, it is instantiated from the scheme template as a behavior into the action selection mechanism (see Figure 1) and allowed to compete for execution. This behavior shares the base-level activation of the scheme which, when aggregated with its current activation, produces a two-factor assessment of the anticipated payoff in selecting this behavior for execution. That is, goal-end motivation and past reliability produce anticipation value such that the satisfaction of deep goal(s) in the future and likelihood of success biases what action is to be executed during the current cycle.

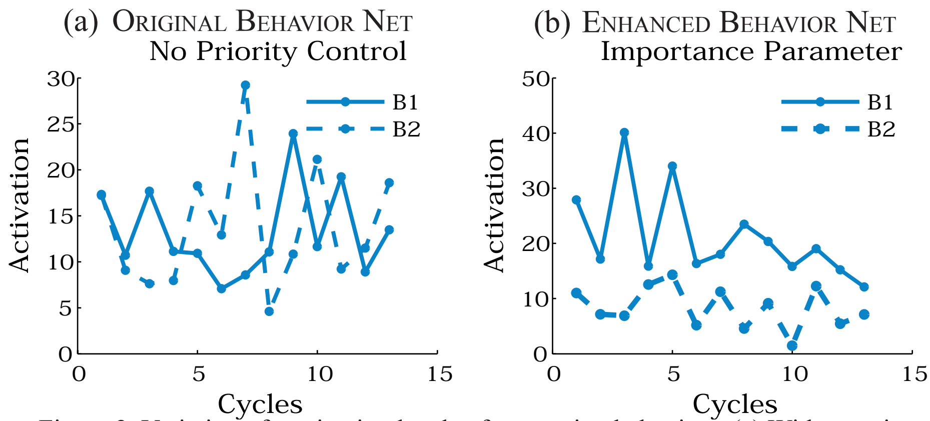

Figure 2. Variation of motivation levels of competing behaviors: (a) Without priority control (Maes' mechanism); (b) With the effect of the importance parameter to control priorities (LIDA's mechanism).

3.2 State Anticipatory Mechanism

In the design of a state anticipatory mechanism we are concerned with explicit predictions of future states influencing current decision making. In LIDA, state anticipations come to play when the agent is confronted by novel situations in which it fails to converge upon an existing plan of action (behavior stream). In these situations, LIDA utilizes its non-routine problem solving (NRPS) process to generate solutions, usually the adaptation of existing ones, to handle an encountered novelty (McCauley, Negatu and Franklin, in preparation; Negatu, 2006. This is on par with the solution finding strategy called meshing (Glenberg, 1997), which in humans, is typically accomplished by putting together bits and pieces of knowledge and techniques that have been stored and, perhaps, used in the past to help solve other problems.

The LIDA's NRPS process starts with novelty detection by using selective attention to notice anomalies in what is expected. Once a novelty is detected, certain attention codelets attempt to recruit relevant resources (building blocks to solutions) in the form of schemes from procedural memory. This is an adaptation of a learning system that follows Edelman's (1987) neuronal group selection mechanism, which itself is inspired by Piaget's (1952) theory of cognitive development. As the meshing process continues, descriptions of intermediate steps or sub-problems (schemes) come to attention which, in turn, recruits more resources from procedural memory.

The above description indicates that the NRPS process guides a controlled partial-order planner. While it shares similarities to dynamic planning systems it differs from earlier approaches such as the general problem solver (Newell, Shaw, & Simon, 1958) in that selective attention is used to target relevant solutions from procedural memory, thus pruning the search space on the basis of the current world model. Without going into the details, similar to any high-level planning system, the NRPS mechanism is a type of animat learning system that makes state anticipations, i.e., planning action decisions are biased towards selecting a plan operator that satisfies a required goal/sub-goal state.

3.3 Sensorial Anticipatory Mechanism

Rather than directly influence the selection of behaviors, sensorial anticipatory mechanisms influence sensorial processing (Butz, Sigaud, & Gerard, 2002). The LIDA system recognizes two forms of sensorial anticipation, the biasing of the senses similar to a preafferent signal (Freeman, 1999) and preparatory attention (LaBerge, 1995).

As described above, nodes of the agent's perceptual associative memory, the slipnet, constitute the agent's perceptual symbols, representing individuals, categories and simple relations. Additionally, schemes in the agent's procedural memory represent uninstantiated actions and action sequences. The context and results of the schemes are represented by the same nodes for objects, categories, and relations in perceptual associative memory. A behavior in the behavior net can be considered an instantiated scheme, thereby sharing its context (as preconditions) and results (as postconditions). Once a behavior is selected in the behavior net, the nodes of the slipnet that compose the postconditions of the behavior have their activations increased, thus biasing them towards selection in the next cycle.

Preparatory attention in LIDA is also implemented on the basis of the currently selected behavior. Each behavior is equipped with one or more expectation codelets, a special type of attention codelet that attempts to bring the results of selected action to attention. Once a behavior is selected for execution, its expectation codelets attempt to bring the results of the behavior to attention, thereby biasing selective attention. In this manner the LIDA system incorporates a second form of action driven sensorial anticipation.

4 Anticipatory Learning

In this section we explore an automatization mechanism to learn low-level implicit anticipations, and a procedural learning mechanism to learn the context and results of existing actions, which in turn, are used to construct a variety of anticipatory links.

4.1 Automatization

Automatization is defined as the ability to learn a procedural task to such an extent that the amount of attention devoted to the routine steps in accomplishing the task is reduced. Automatization develops as procedural tasks are rehearsed with attentional intervention. However, once tasks have been sufficiently automatized, the cognitive processing shifts from serial, attentional, and controlled to parallel, non-attentional, automatic (Logan, 1980). In LIDA a procedure produces a stream of actions with the execution of each action furthering the progression from the current state towards some goal state. For an unrehearsed procedure the transitions between consecutive actions are usually controlled by selective attention. The automatization process forms associations between unattended, coactive, low-level processes (behavior and attention codelets). As the associations between these processes strengthen over time (rehearsal) the need for selective attention gradually fades.

A complete description of the automatization mechanism is beyond the scope of this paper (see Negatu, 2006; Negatu, McCauley, & Franklin, in review). However, Figure 3(a) illustrates the basic automatization process by depicting an execution sequence of a task at a high level (the behavioral level, B1 and B2) before automatization, and at a low level (the codelet level, BC1, BC2, AC1, AC2). In the absence of automatization selective attention is needed to activate the successor link between the high level behaviors B1 and B2. However, over time a number of associations between the low-level codelets are formed. When the associative link between BC1 and BC2 is sufficiently strong the link between behavior B1 and B2 can be implicitly activated without any attentional control, that is, without B2 being selected by the action selection mechanism.

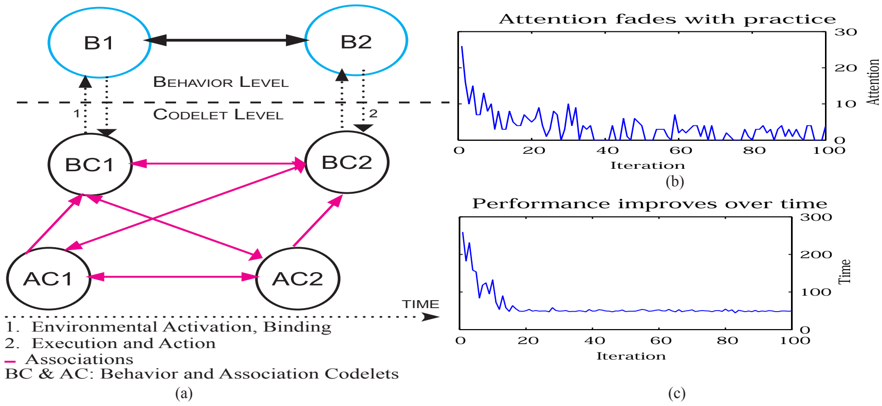

Figure 3. Automatization mechanism with experimental results

Details of the implementation and experiments of the automatization mechanism can be found in Negatu (2006). In order to describe the effects of the automatization process we setup a simple experiment to perform a walking task that required a sequence of actions to execute. Figures (3b) and (3c) correspondingly show that attention fades and performance improves as the degree of automatization increases with each iteration until it saturates.

The automatization mechanism implicitly causes a controlled task execution process to transition into a highly coordinated skill thus improving performance and reserving attention, a limited resource, for more novel tasks. It is a type of implicit anticipatory learning mechanism since the encoding of the experiences of performing tasks is integrated in, and arises from, the payoff anticipatory process of LIDA's action selection dynamics. Failures or unexpected outcomes during the execution of automatized tasks require deautomatization, i.e., the reengagement of attention (Negatu, 2006).

4.2 Procedural learning

The discussion so far has assumed that procedural memory in LIDA is built in and does not learn. In such situations, the use of anticipations is restricted to what was engineered into the system, thus greatly restricting its scope. In order to alleviate this problem we briefly describe a procedural learning mechanism in which the context and results of action schemes are learnt.

The procedural learning mechanism is a variant of Drescher's (1991) schema mechanism and is based on selective attention and reinforcement learning. Reinforcement is provided via a sigmoid function such that initially reinforcement increases very rapidly but tends to saturate. By negating and translating the same sigmoid curve by +1 we obtain the decay curve. Therefore, schemes with low base-level activation (measure of reliability, used to determine payoff) decay rapidly, while schemes with high (saturated) base level activation values tend to decay at a much lower rate.

For learning to proceed initially, the behavior network must first select the instantiation of an empty scheme for execution. Before executing its action, the instantiated scheme spawns a new expectation codelet. After the action is executed, this newly created expectation codelet focuses on changes in the environment that result from the action being executed, and attempts to bring this information to attention. If successful, a new scheme is created, if needed. If one already exists, it is appropriately reinforced. Perceptual information selected by attention just before and after the action was executed form the context and result of the new scheme respectively. The scheme is provided with some base-level activation, and it is connected to its parent empty scheme with a link. More details on this mechanism can be found in (D'Mello et al., 2006).

Of importance to this paper are the effects that the learning of a new scheme have on the anticipatory processes. The creation of a new scheme leads to a number of new anticipatory links being formed. The result of the scheme can be used to learn new expectation codelets to monitor future execution. These expectation codelets can be used to assess the reliability of this scheme thus influencing payoff anticipations. They also serve as sensorial anticipations by biasing perceptual associative memory and selective attention.

5 Discussion

This paper has described the different anticipatory aspects of the primary mechanisms that comprise the LIDA architecture. We have described the manner in which payoff anticipations are realized in LIDA through a drive based motivational system and a reinforcement learning mechanism. It should be noted that the use of a drive based motivation scheme in assessing the payoff in selecting a behavior may not clearly fit into one of the suggested distinctions of payoff vs. state anticipation. It has been suggested that such motivations and/or emotion systems, in influencing action decisions, indirectly predict states. Thus it could be argued that in reality these systems constitute a type of state anticipation. Sensorial anticipations occur through a preafferent signal (Freeman, 1999) and preparatory attention (LaBerge, 1995). LIDA makes explicit predictions of future state while attempting to solve novel problems thereby realizing a form of state anticipation. One could also argue that implicit anticipatory mechanisms are realized in LIDA by the individual behaviors in the behavior network.

Using the automatization mechanism described above, implicitly anticipatory links among the low-level processors (codelets) are learnt on the basis of anticipatory (payoff) decision making by the high level constructs (behaviors). The procedural learning mechanism while forming new schemes from the causal relationships between the context and results of actions creates expectation codelets that in turn perform various anticipatory tasks. Although our model of procedural learning is motivated by Drescher's schema mechanism (1991), our learning mechanism differs in that selective attention assumed to be a necessary condition for supraliminal learning. While learning in Drescher's system relies on each schema maintaining several reliability statistics, we only use a single, computationally more tractable statistic, the base-level activation modeled by a saturating sigmoid function.

In an introductory paper, Butz et al., (2002) provide examples of a number of artificial systems that incorporate anticipations at some level. The major difference between LIDA's anticipatory mechanisms and those of other systems is that LIDA attempts to solve non-Markov problems in that it does not assume that the current sensorial input is sufficient for decision making. Blank et al., (2005) emphasize the merits of connectionist anticipatory systems with low-level representations and the limitations of discrete and symbolic anticipatory systems such as Witkowski's system (2002). Incorporation of connectionist flavored mechanisms (perceptual associative memory and action selection) and low-level representation to encode anticipatory learning (e.g. automatization mechanism) give added merit to LIDA as an animat system.

In this paper we have demonstrated that a wide variety of anticipatory mechanisms can be quite naturally integrated into a broad, comprehensive, autonomous agent architecture modeled after human cognition. These anticipatory mechanisms play important roles in perceiving, attending and action selection. Robots and software controlled by such an agent architecture could be expected to perform better as a result of these anticipatory mechanisms. The inclusion of anticipatory mechanisms in the LIDA architecture may well serve to model their inclusion in other such architectures.

References

1. Baars, B. J. 1988. *A Cognitive Theory of Consciousness*. Cambridge.: Cambridge University Press.
2. Barsalou, L. W. 1999. Perceptual symbol systems. *Behavioral and Brain Sciences* 22:577–609.
3. Blank, D. S., Lewis, J. M., and Marshall, J. B. 2005. The Multiple Roles of Anticipation in Developmental Robotics. AAAI Fall Symposium Workshop Notes, *From Reactive to Anticipatory Cognitive Embodied Systems*. AAAI Press
4. Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*. RA-2:14-23.
5. Butz, M. V.; Sigaud, O.; and Gerard, P. 2002. Internal models and anticipations in adaptive learning systems. In *Proceedings of the Workshop on Adaptive Behavior in Anticipatory Learning Systems*, 1–23.
6. Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
7. D'Mello, S. K., Franklin, S., Ramamurthy, U., & Baars, B. J. 2006. A Cognitive Science Based Machine Learning Architecture. AAAI Spring Symposia Technical Series, Stanford CA, USA. Technical Report SS-06-02 (pp. 40-45). AAAI Press
8. Drescher, G. 1991. *Made Up Minds: A Constructivist Approach to Artificial Intelligence*, Cambridge, MA: MIT Press.
9. Edelman, G. 1987. *Neural Darwinism: the theory of neuronal group selection*, Basic Books.
10. Franklin, S. 2001. Automating Human Information Agents. *Practical Applications of Intelligent Agents*, ed. Z. Chen, and L. C. Jain. Berlin: Springer-Verlag.
11. Franklin, S., Baars, B.J., Ramamurthy, U., and Ventura, M. 2005. The Role of Consciousness in Memory. Brains, Minds and Media, Vol.1, bmm150 (urn:nbn:de:0009-3-1505)
12. Freeman, W. J. 1999. *How Brains Make Up Their Minds*. London: Weidenfeld & Nicolson General.
13. A. Glenberg 1997. *What memory is* for" Behavioral and Brain Sciences 20, pp1-55.

14. Hofstadter, D. R., and M. Mitchell. 1994. The Copycat Project: A model of mental fluidity and analogy-making. In *Advances in connectionist and neural computation theory*, Vol. 2: logical connections, ed. K. J. Holyoak, and J. A. Barnden. Norwood N.J.: Ablex.
15. Kay, L.M. and Freeman, W.J. 1998. Bidirectional processing in the olfactory-limbic axis during olfactory behavior. *Behavioral Neuroscience*, 112(3): 541-553
16. Kunde, W. 2001. Response–effect compatibility in manual choice reaction tasks. *Journal of Experimental Psychology: Human Perception & Performance*, 27, 387-394.
17. LaBerge, D. 1995. *Attentional processing: the brain's art of mindfulness*. Harvard University Press, 1995
18. Logan, G. D. 1980. Attention and automaticity in Stroop and priming tasks: Theory and data. *Cognitive Psychology, 12,* 523-553
19. Maes, P. 1989. How to do the right thing. *Connection Science* 1:291-323.
20. McCauley, L., A. Negatu, and S. Franklin. in preparation. An Agent Architecture for Non-Routine Problem Solving.
21. Negatu, Aregahegn, 2006. Cognitively Inspired Decision Making for Software Agents: Integrated Mechanisms for Action Selection, Expectation, Automatization and Non-Routine Problem Solving, Ph.D. Dissertation, The University of Memphis, USA.
22. Negatu, A., T. L. McCauley, and S. Franklin. in review. Automatization for Software Agents.
23. Newell, A., Shaw, J. C., & Simon, H. A. 1958. Elements of a theory of human problem solving. *Psychological Review*, 65 , 151-166.
24. Piaget, J. 1952. *The Origins of Intelligence in Children*. New York: International Universities Press.
25. Rosen, R. : *Anticipatory Systems*. Pergamon Press, 1985
26. Schubotz, R.I., von Cramon, D.Y. 2001 *Functional organization of the lateral premotor cortex*. fMRI reveals different regions activated by anticipation of object properties, location and speed. Cognitive Brain Research 11 97-112
27. Stolzmann, W. 1998. *Anticipatory Classifier Systems*. in Genetic Programming. University of Wisconsin, Madison, Wisconsin: Morgan Kaufmann
28. Witkowski, M. 1997 *Schemes for Learning and Behaviour: A New Expectancy Model*, Ph.D. thesis, University of London.