An Autonomous Agent Architecture for Integrating "Unconscious" and "Conscious", Reasoned Behaviors*

Henry H. Hexmoor, Johan M. Lammens & Stuart C. Shapiro

Department of Computer Science
SUNY at Buffalo, Buffalo NY 14260
e-mail {hexmoor|lammens|shapiro}@cs.buffalo.edu
(To appear in proc. Computer Architectures for Machine Perception 93)

1 INTRODUCTION

In contrast to "conscious", reasoned behaviors, we consider behaviors that are automatic and unreasoned to be "unconscious". The latter are commonly found in behavior-based AI models [Bro90, Mae90]. We are developing an architecture that models agents with both "conscious" and "unconscious" behaviors. Furthermore, we are interested in modeling agents that learn behaviors from their successful interactions with the world. We call these learned behaviors "emergent behaviors". We present an architecture for intelligent autonomous agents which we call GLAIR (Grounded Layered Architecture with Integrated Reasoning) [HLS92, HCBS93, HLS93, HLCS93, LHS]. GLAIR is a general multi-level architecture for autonomous agents with sensory and motor capabilities. GLAIR offers "unconscious" layers for modeling tasks that exhibit a close affinity between sensing and acting, i.e., behavior-based AI modules, and a "conscious" layer for modeling tasks that exhibit delays between sensing and acting, and require deliberation on the part of the agent. GLAIR provides learning mechanisms that allow for autonomous agents to learn emergent behaviors and add them to their repertoire of behaviors. In this paper we will describe the principles of GLAIR, and an application we have developed that demonstrates how GLAIR-based agents acquire and exhibit a repertoire of behaviors at different cognitive levels.

2 OVERVIEW OF GLAIR

What goes into an architecture for an autonomous agent has traditionally depended to a large extent on whether we are *building a physical system*, *understanding/modeling behaviors of an anthropomorphic agent*, or *integrating a select number of intelligent behaviors*. The organization of an architecture is also influenced by adopting various philosophical positions like Fodor's *modularity* assumption [Fod83], or a *connectionist* point of view, e.g. [MRH86], or an *anti-modularity* assumption as in Brooks's subsumption architecture [Bro85]. The *modularity* assumption supports (among other things) a division of the mind into a *central system*, i.e. cognitive processes such as learning, planning, and reasoning, and a *peripheral system*, i.e. sensory and motor processing [Cha90]. Our architecture is characterized by a three-level organization into a Knowledge Level (KL), a Perceptuo-Motor Level (PML), and a Sensori-Actuator Level (SAL). This organization is neither modular, anti-modular, hierarchical, anti-hierarchical, nor connectionist in the conventional sense. It integrates a traditional symbol system with a physically grounded system, i.e. with a *behavior-based* architecture. The most important difference with a purely behavior-based architecture like Brooks's subsumption architecture is the presence of three distinct levels with different representations and implementation mechanisms for each, particularly the presence of an explicit KL. Representation, reasoning (including planning), perception, and generation of behavior are distributed through all three levels. Our architecture is best described using a resolution pyramid metaphor as used in computer vision work [BB82], rather than a central vs. peripheral metaphor. Architectures for building physical systems, e.g. robotic architectures [ABN81], tend to address the relationship between a physical entity (like a robot) and sensors, effectors, and tasks to be accomplished. Since these physical systems are performance centered, they often lack general knowledge representation and reasoning techniques. These architectures tend to be primarily concerned with the *body*, that is, how to get the physical system to exhibit intelligent behavior through its physical activity. These architectures address what John Pollock calls *Quick and Inflexible* (Q&I) pro-

*This work was supported in part by Equipment Grant No. EDUD-US-932022 from SUN Microsystems Computer Corporation.

cesses [Pol89].

Architectures for understanding/modeling behaviors of an anthropomorphic agent, e.g., cognitive architectures [And83, Pol89, LMA91], tend to address the relationships that exist among the structure of memory, reasoning abilities, intelligent behavior, and mental states and experiences. These architectures often do not take the *body* into account. Instead they primarily focus on the *mind* and *consciousness*. We operationally define consciousness for a robotic agent as being aware of one's environment, as evidenced by (1) having some internal states or representations that are causally connected to the environment through perception and action, (2) being able to reason explicitly about the environment, and (3) being able to communicate with an external agent about the environment. Our architecture ranges from general knowledge representation and reasoning to body-dependent physical behavior, and the other way around. We are interested in autonomous agents that are embedded in a dynamic environment. Such an agent needs to continually interact with and react to its environment and exhibit intelligent behavior through its physical activity. To be successful, the agent needs to reason about events and actions in the abstract as well as in concrete terms. This means combining situated activity with acts based on reasoning about goal-accomplishment, i.e., deliberative acting or planning. In the latter part of this paper, we will present a particular agent based on our architecture. This agents is designed with a robot in mind, but its structure is also akin to anthropomorphic agents. Figure 1 schematically presents our architecture. There are several features that contribute to its robustness. We highlight them below. For an in-depth discussion and comparison with other architectures see [HLS92, LHS].

- We differentiate conscious reasoning from unconscious Perceptuo-Motor and Sensori-Actuator processing.
- The levels of our architecture are semi-autonomous and processed in parallel.
- Conscious reasoning takes place through explicit knowledge representation and reasoning. Unconscious behavior makes use of several different mechanisms.
- Conscious reasoning guides the unconscious behavior, and the unconscious levels, which are constantly engaged in perceptual and motor processing, can *alarm* the conscious level of important events, taking control if necessary. Control and generation of behavior are layered and not exclusively top-down.
- Lower level mechanisms can preempt higher level ones (this is almost subsumption [Bro85] on its head).
- There is a correspondence between terms in the Knowledge Representation and Reasoning (KRR) system on one hand, and sensory perceived objects, properties, events, and states of affairs in the world and motor capabilities on the other hand. We call this correspondence *alignment*.
- The level at which any given behavior is generated and/or controlled is not fixed, but can vary in the course of learning, or depending on the particular goals and capabilities of the agent in question.

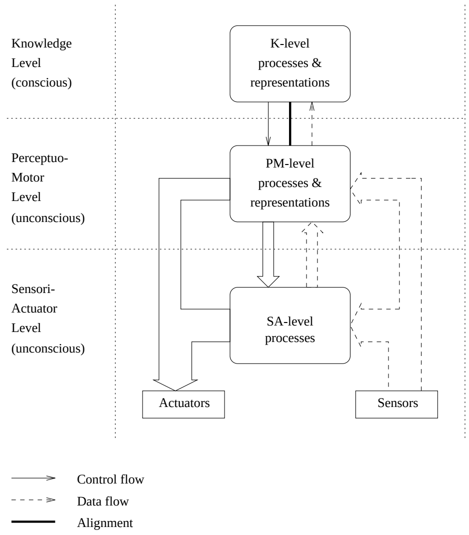

Figure 1. Schematic representation of the agent architecture. Width of control and data paths suggests the amount of information passing through (bandwidth). Sensors include both world-sensors and proprio-sensors.

One major objective for GLAIR is learning emergent behaviors. In accordance with Agre's improvised actions [AC87] and Brooks's subsumption [Bro85] we believe that complex behaviors emerge from interaction of the agent with its environment without planning. However, previous work in this area hard-coded primitive actions and did not attempt to learn the improvised behavior. We include mechanisms in our architecture for detecting emergent behavior at the unconscious layers and study how corresponding concepts can be learned at the conscious layer. As an example of the interaction between reflexes and other types of behavior, consider a mobile robot going down a hall. Keeping to the middle of the hall is implemented at the PML, while obstacle avoidance is a reflex at the SAL. In case the robot comes too close to an obstacle, the reflex takes control, stops the robot, makes it retreat some, and changes its orientation slightly. Simultaneously a signal is sent to the PML and the KL to signify that a reflex has occurred. A good choice of reflex behavior (the retreat and orientation change in this case) will in most cases result in obstacle avoidance even if the reflex is ignored (after repeated trial and error, somewhat like a mechanical toy with a simple sensor avoids falling off the edge of a table). Compared to Brooks's approach, the main difference is that the subsumed behavior (forward motion) is notified of the event, and may or may not take that into account subsequently. This is more psychologically realistic in our opinion, as well as a more flexible control strategy.

Another major objective for GLAIR is to study how symbolic Knowledge Level concepts can be grounded in perception and action, giving these concepts an *embodied semantics* for the robotic agent. We believe that if one wants to use symbolic modeling as part of an autonomous agent architecture, one has to make sure the symbols in those models are meaningful *to the agent*, rather than to the designer of the system, and to insure this, we need a set of grounded or embodied symbols that can function as a basis of meaning for all symbols in the model. We define embodiment as the meaning or extension of high level symbols being determined both by the agent's own physiology (bodily functions) and by its interactions with the world [HLS93]). For instance, a KL symbol "grasp" which is aligned with a PMA for grasping has an embodied semantics, but a symbol "fly" for which there is no alignment is not. A KL symbol "red" which is aligned with a particular region in the agent's perceptual color space at the PM level is embodied, but a symbol "beautiful" for which there is no alignment is not. Embodied symbols are intrinsically meaningful to the agent, non-embodied ones can only be by virtue of being systematically related to embodied ones. The latter kind of symbols we consider to be indirectly grounded, after [Har90].

2.1 Perceptuo-Motor Automata

At the perceptuo-motor level, the behaviors resulting in physical actions are generated by an automaton, which we will call a PM-automaton (PMA) [HN92, HLCS93]. In other words, PMAs are representation mechanisms for generating behaviors at an "unconscious" level. PMAs are finite state machines represented by <Rewards, Goal-Transitions, Goals, Action-Transitions, Actions, Sensations>. Goals, Rewards, and Goal-Transitions in a PMA can be empty. The primary mode of acquiring a PMA is intended to be by converting plans in the knowledge level into a PMA by a process described in [HN92]. In Section 3, we describe a stepwise learning scheme for acquiring PMA transitions.

2.2 Color Perception

As a particular instance of symbol grounding and embodied symbolic concepts, we have been working on a model of color perception and color naming, which will provide grounding for some of the symbolic representations at GLAIR agents' Knowledge level. This model allows an agent to (1) name colors shown to it, and express a typicality judgement, (2) point out examples of named colors in its environment, and (3) learn new names for colors. This model provides the perceptual grounding for a set of basic color terms [BK69] represented at the Knowledge Level. The color domain was chosen as a case study for embodied perception because of the relative abundance of psychological, psychophysical and neurophysiological data in the literature. It is a complex enough area to allow the usefulness of embodiment for computational models of perception to be demonstrated, yet feasible enough to be implemented in actual autonomous agents. Our research draws on work in the neurophysiology of color perception, particularly [DAJ66], in semantic universals for natural languages, particularly [BK69], and other work in AI and cognitive science.

At the most general level of description, our model has to explain (and reproduce) a signal-to-symbol transition, going from light entering a sensing device (camera) to symbols representing the perceived color at any point in the image. To make our problem manageable, we make some simplifying assumptions. We are only concerned with single-point determination of color, thus disregarding spatial interactions in color perception. We also assume foveal cone photoreceptors as our neurophysiological reference. Furthermore, we restrict the problem to any given fixed state of adaptation of the vision system, thus avoiding issues of color constancy [Boy79, WS82]. In the physical agent implementations we are developing we will deal with these issues to some extent, using a scaling technique. Conceptually, we break the model up into two parts: the first part which takes us from the visual stimulus (SA level) to color space coordinates (PM level), the second from color space coordinates to a set of color names (K level).

Since we hypothesize that the nature of the color perception mechanism underlying human color naming determines to some extent the existence and the nature of the semantic universals of color, we want to use a color space which is based on the neurophysiology of color perception. Based on data published in [DAJ66], we have reconstructed 3D models of the response of 4 types of color-opponent cells and 2 types of non-opponent cells in the Macaque LGN. Response is a function of both spectral composition and radiance of the stimulus. By pairwise combining these functions, we have derived new 3D response functions that we take as the basis of an opponent color space: Green-Red and Blue-Yellow opponent functions and a non-opponent Brightness function (Fig. 2). We will refer to this newly defined space as the Valois color space. We have reasons to believe that existing opponent color models as used in computer vision and computer graphics (for instance [JH55]) are not accurate as models of color perception, but we will not discuss that here. Most work in Color Science [WS82] and color computer vision is based on the CIE XYZ color space and derivatives thereof, so we want a transformation from XYZ coordinates to Valois space. The transformation we are experimenting with reduces to an equation of the form

$$\mathbf{v} = s(\mathbf{M_2} \cdot s(\mathbf{M_1} \cdot \mathbf{i})) \tag{1}$$

where $\mathbf{v}$ represents the 3×1 matrix of Valois coordinates, $\mathbf{M}_j$ represents a 3×3 matrix for stage j, $\mathbf{i}$ represents the 3×1 matrix of XYZ input coordinates, and s represents the sigmoid activation function commonly found in artificial neural networks:

$$s(x) = \frac{1}{1 + e^{(\frac{h-x}{t})}} \tag{2}$$

with $x \in \mathcal{R}$ (the input), $h \in \mathcal{R}$ (the half-response input), and $t \in [0, 1]$ (the "temperature", determining the steepness of the curve). Figure 3 shows the optimal color stimuli solid [WS82] in different color spaces, including a preliminary version of the Valois space. Note the different shapes of the solid, and how it is "warped" in the L*a*b* and Valois spaces. In the Valois space, the shape is not unlike that of the Ostwald and Munsell color spaces (which are psychological ones, based on observation rather than measurement or explicit mathematical models), but this remains to be investigated in detail. We have derived tentative measures for the psychophysical variables of hue, saturation, and brightness from the Valois space.

For The second part of our model, relating Valois coordinates to a set of color names, we need to partition the 3D Valois space into a set of volumes or regions, each corresponding to a named color concept (or category). From a model-theoretical point of view, we can consider these regions to be the extensional referents of the symbols representing color concepts. From the work of [BK69] and others we know that there exists a set of semantic universals in the domain of color, known as basic color categories (BCC), each corresponding to a "basic color term" (BCT). We want our naming algorithm to learn the BCTs of a set of different languages (one at the time) corresponding to (a subset of) the universal BCCs. Depending on the requirements we impose on the shape of the BCC regions of our color space, we can create binary valued or continuous valued "characteristic functions" for the regions, and have the regions overlap or not. We could interpret the BCCs as fuzzy sets [Zad71, KM78], for instance. From the work of [BK69] and others we know that human BCC characteristic functions are continuous (or at least non-binary) valued, and that the regions may or may not overlap, perhaps depending on the particular experimental paradigm used.

2.3 Knowledge Migration

Knowledge in GLAIR can migrate from conscious to unconscious levels. In our application Air Battle Simulation (Section 3) we show how a video-game playing agent learns how to dynamically "compile" a game playing strategy that is initially formulated as explicit reasoning rules at the Knowledge level into an implicit form of knowledge at the Perceptuo-Motor level, a PMA. When the knowledge level determines an action for execution, the action reaches the PM-level. At the PM-level, this knowledge is learned in the form of PMA transitions. The idea is that the next time circumstances make this action applicable, it can be selected for execution without having to resort to "conscious" processes.

2.4 Learning Tendencies

An agent that is able to do one of several things in a given situation, should choose to do what in the long run benefits it most. We are interested in how an agent can improve its skills by improving its choice of actions. Our premise is that an agent has some basic skills but does not always choose the best possible action. We are concerned with unconscious choices of action, i.e. at the PM level.

Reinforcement based learning, particularly Q learning [Wat89], is a successful technique that has been used for learning action consequences, also known as

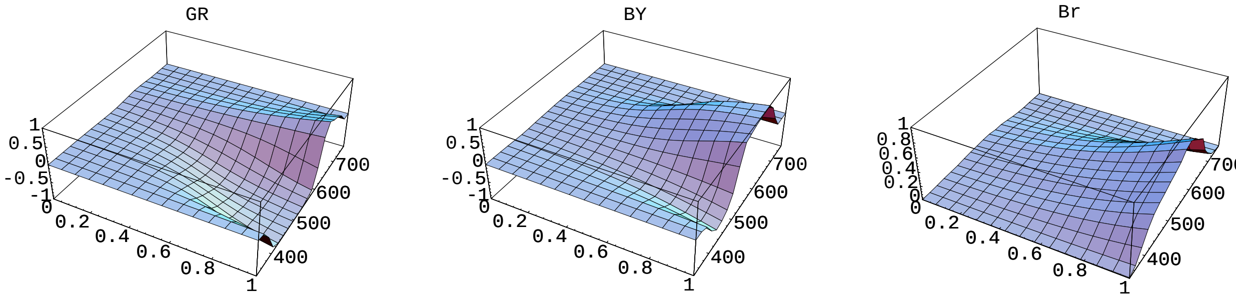

Figure 2. Derived basis functions for the Valois color space. From left to right: the Green-Red and Blue-Yellow opponent functions, and the Brightness (Dark-Light) non-opponent function. On the X axis: Input radiance (I) in log units relative to the threshold response level; Y axis: wavelength (L) in nm; Z axis: activation (A) in $[-1, 1]$ normalized units.

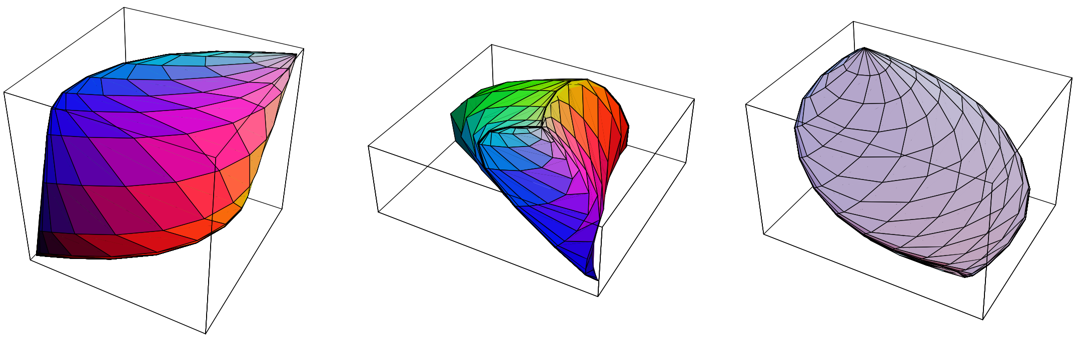

Figure 3. The optimal color stimuli (OCS) solid in CIE XYZ space (left), transformed into CIE L*a*b* space (middle) and a preliminary version of the Valois space (right). On the X,Y,Z axes (arbitrary units): CIE X,Y,Z coordinates; CIE a*,b*,L* coordinates; Valois Green-Red,Blue-Yellow,Brightness coordinates.

action models.[1] In learning action models, a Markovian environment is assumed. That is, the agent believes the world changes only due to its own actions. In contrast to learning action models, we are interested in modeling behavior generation by agents that function in dynamic environments. The impact of the agent's action on itself depends on the situations under which actions are applied and on other agents' actions. Other agents' actions are assumed to be nondeterministic. Furthermore, we assume that the agent is computationally and cognitively resource bounded. We assume that the agent needs time to think about the best action, and in general there is not enough time. In such an environment, we want the agent to observe its own interactions with the world in order to learn what action tends to be successful in light of its goals and prevailing situations. This is useful when the agent has to arbitrate among multiple applicable actions.

[1] In Q learning, **Utility**(S(t),C(t)) is the evaluation of how appropriate command C is in situation S when C is executed at time t in response to S at t. **R**(S(t+1)) is the reward received by being in state S(t+1). For the ABS agent (Section 3), rewards are determined as the game is being played and cannot be determined beforehand. This is called the immediate reward. The following equality is maintained and propagated as each command is executed. At the start of the game, all **Utility**(S,C) in the PMA are set to 1.
Utility(S(t), C(t)) = **R**(S(t+1)) + γ [λ **Utility**(S(t+1), C(t+1)) + (1 - λ) $\max_{C(t+1)}$**Utility**(S(t+1), C(t+1))]
γ is a parameter to determine how important it is to be in the state that the pilot ends up in after his move. In reinforcement based learning this is known as the discount factor. λ is known as the recency/learning factor.

2.5 Learning Emergent Routines

We endow the agent with a minimal number of primitive actions and sensations. Our basis of this minimality and choice of primitive actions is physiological. In other words, in our modeling a computer agent, we will choose actions that are physically basic for the robot as primitive actions for the computer agent. We then instruct the agent to perform tasks and in the midst of accomplishing them, we expect the agent to notice basic behaviors (i.e., routines) emerge. An example of an emergent behavior we will explore is when a mobile robot learns to *move toward an object*. In our application Mobile Robot Lab (MRL) [HLCS93] we discuss how a mobile robot agent might learn emergent behaviors and consequently use them in its repertoire of actions. We assume that the agent does not know about the long term consequences of its actions. Over a finite number of actions, when the agent observes a substantially improved situation, chances are it has found a successful *routine*. We record such detected *routines* and as they reoccur, we increase our confidence in them. When our confidence in a *routine* reaches a certain level, a concept is created at the Knowledge level of GLAIR for the routine and from then on, this routine can be treated as a single action at that level. Learning routines is closely related to a variation of learning tendencies as we discussed in the previous section. Instead of learning action goodness, to learn a routine we record the sequence of actions between a significantly bad situation and a significantly good situation.

3 AN APPLICATION: AIR BATTLE SIMULATION

We have developed a World War I style dogfight simulation game we call Air Battle Simulation (ABS). Figure 4 schematically presents the structure of the GLAIR agent that plays the Air Battle Simulation video-game. We will refer to this agent as "Gabby," for "GLAIR air battler." Initially, Gabby has not acquired a PMA for the game yet, and so uses conscious level reasoning (i.e., SNePS behavioral rules [SR87]) to decide what move to make. Once transitions are learned and cached in a PMA, Gabby uses the PMA for deciding its next move whenever possible. By adding learning strategies, a PMA can be developed that caches moves decided at the knowledge level for future use. Here again, learning can be used to mark PMA moves that prove unwise and to reinforce moves that turn out to be successful. We are exploring these learning issues. Gabby particularly demonstrates real time behaviors and the inter-level

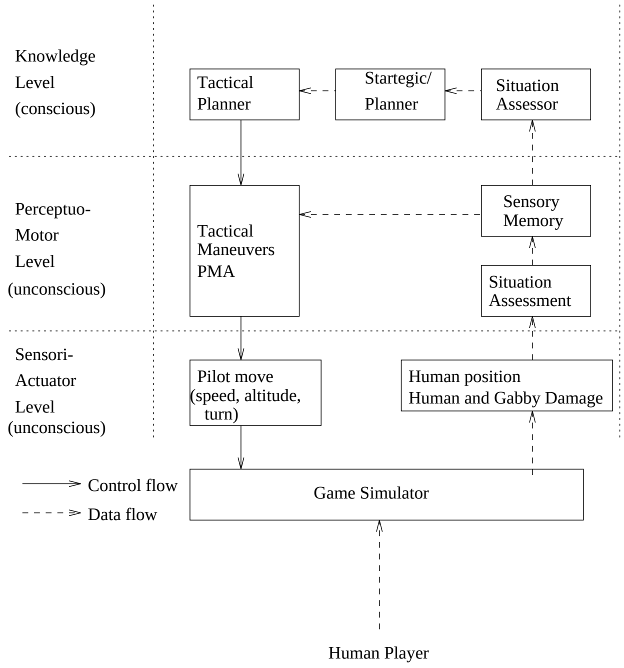

Figure 4. Structure of the ABS agent.

alignment mechanism.

The ABS video game simulates dog-fights between Gabby and a human player. The game display consisting of a main window giving a horizontal view, a side window giving a vertical view, two "damage reports" (see Figure 5), and a control panel window, (see Figure 6). The human player's plane is always considered to be in the center of the two shaded areas. The horizontal two-dimensional position and orientation of Gabby's plane is displayed by the triangle on the main window, and its height relative to the human's plane is indicated by the drawing of a plane in the side window. The condition of the human's plane is indicated by the report labelled "Own Damage," and the condition of Gabby's plane by the report labeled "Enemy's Damage." When the two planes are close in all three dimensions, as indicated by Gabby's plane being shown in the two shaded areas, whichever plane is facing the other one automatically fires. (That is, neither Gabby nor the human makes a separate decision about when to fire.)

The human player uses the control panel to choose a move, which is a combination of changing altitude, speed, and direction. When the human player presses the GO button, Gabby also selects a move. The game simulator then considers the two moves to determine the outcome, and updates the screen and the accumulated damage to the two planes, thus simulating simultaneous moves. The game ends when one or both of the two player's planes are destroyed.

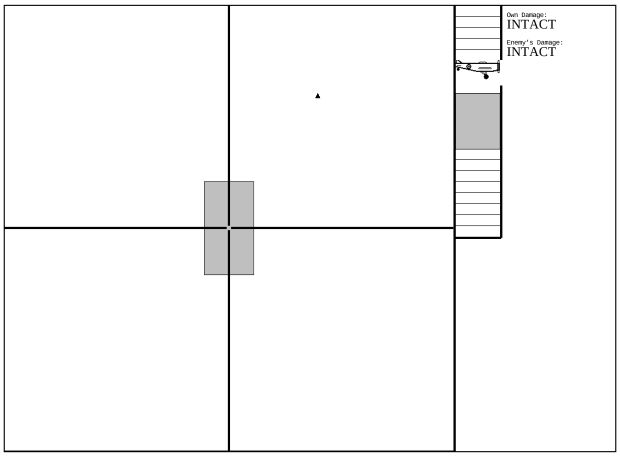

Figure 5. Air Battle Simulation game windows. Gabby's plane is indicated by the small triangle in the upper right quadrant of the main window, and by the drawing of a plane in the side window. This figure shows Gabby fleeing, flying parallel, and at a higher altitude than the human. The shaded regions denote shooting range. If Gabby's plane appears in both shaded regions, whichever plane is facing the other (possibly both) will fire.

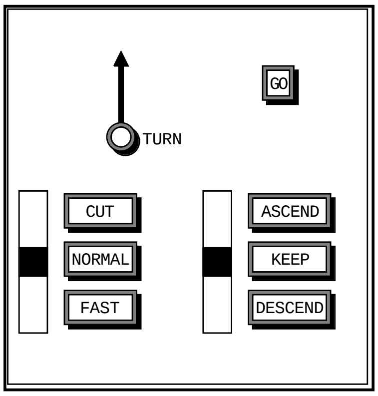

Figure 6. Air Battle Instrument Panel window. To select/change a move, the human player pushes one of the buttons in each column and adjusts the TURN dial. CUT, NORMAL, and FAST are used for speed selection. ASCEND, KEEP, DESCEND are used for altitude change. Pushing the GO button submits the move to the game simulator.

3.1 Knowledge Migration in ABS

In ABS, when the knowledge level determines an action, it submits a pair consisting of an action A and a goal G to the PM-level. At the PM-level of ABS, goals are one-to-one associated with behaviors. Similarly, actions are one-to-one associated with commands. Let's consider C and B to be the unique counterparts of A and G, respectively.

If there is no command transition corresponding to the triple <S, B, C> (with S representing the current situation), a new command transition CT is learned. CT is then S $\times$ current goal $\mapsto$ C. If G is different from the previous goal submitted to PMA, a new behavior transition BT is learned. Let the current situation in PMA be S. BT is then S $\times$ current behavior $\mapsto$ B.

We started ABS with an empty PMA and as the game was played, transitions of the PMA were learned. Figure 7 shows typical ABS transitions learned. As the transitions were learned, when similar situations occurred and there was an appropriate PMA response, the PMA executed that command. As the game was played, we observed that the agent became more reactive since the PMA was used to generate behaviors instead of the knowledge level. Figure 7 shows a small sample of learned PMA transitions while playing ABS.

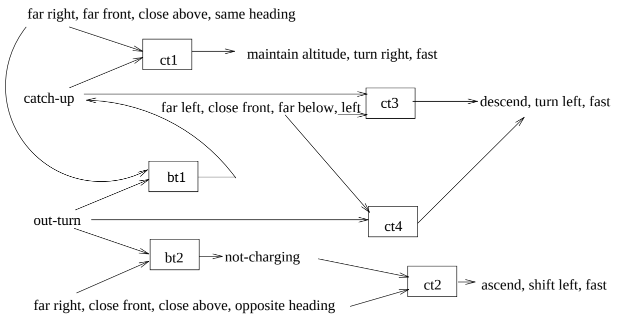

Figure 7. Learned Command and Behavior Transitions

3.2 Learning Tendencies in ABS

The rules of a PMA in ABS are pairs of situation/command. As it turns out, a situation can be paired up with multiple commands. The object of learning here is to learn which commands when associated with a situation yield a better result, i.e., the pilot ends up in a more desirable situation.

Some situations in ABS are more desirable for the pilot than others, e.g., being right behind the enemy and in shooting range. We assign a reward value to each situation S between -1 and 1, $\mathbf{R}(s)$. As the pilot makes a move, it finds itself in a new situation.

This new situation is not known to the pilot beforehand since it also depends on the other pilot's move. Since the new situation is not uniquely determined by the pilot's move, the pilot's view of the game is not Markovian.

To test reinforcement based learning in ABS, we developed a programmed opponent for Gabby that plays according to a fixed strategy, written to simulate the strategy a human player might use. We had the two programs play 25 games without learning. Gabby won about 30% of these games. We then turned on Gabby's reinforcement based learning, and had the programs play an additional 197 games. After the games, Gabby started winning 50% of the time, and didn't improve any further. Figure 8 shows the improvement in Gabby's game playing ability.

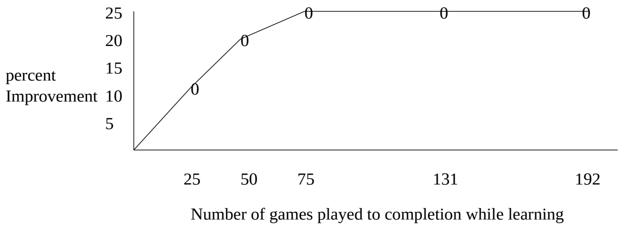

Figure 8. Learning using reinforcement based Q learning

4 SUMMARY AND CONCLUSION

In this paper we have outlined an autonomous agent architecture and one of its instantiations. Our architecture models agents with "conscious" and "unconscious" behaviors. The architecture provides for grounded symbolic representations through embodied perception, and provides a mechanism for learning behaviors. We discussed how the Air Battle Simulation implements an autonomous agent conforming to the architecture, and how knowledge migration and various other features of the architecture apply to it.

REFERENCES

[ABN81] James Albus, Anthony Barbera, and Roger Nagel. Theory and practice of hierarchical control. In *23rd International IEEE Computer Society Conference*, pages 18–38, 1981.

[AC87] Philip E. Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of AAAI-87, Seattle, Wa.*, pages 268–272, July 1987.

[And83] J. R. Anderson. *The Architecture of Cognition.* Cambridge: Harvard University Press, 1983.

[BB82] Dana H. Ballard and Christopher M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs NJ, 1982.

[BK69] Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution.* University of California Press, Berkeley CA, first paperback 1991 edition, 1969.

[Boy79] Rober M. Boynton. *Human Color Vision.* Holt, Rinehart and Winston, 1979.

[Bro85] Rodney Brooks. A robust layered control system for a mobile robot. Technical Report 864, MIT AI Labs, MIT, 1985.

[Bro90] Rodney A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15, 1990.

[Cha90] David Chapman. Vision, instruction, and action. Technical Report 1204, MIT Artificial Intelligence Laboratory, MIT, 1990.

[DAJ66] Russell L. De Valois, Israel Abramov, and Gerald H. Jacobs. Analysis of response patterns of lgn cells. *Journal of the Optical Society of America*, 56(7):966–977, July 1966.

[Fod83] Jerry Fodor. *The Modularity of Mind.* MIT Press, 1983.

[Har90] Stevan Harnad. The symbol grounding problem. *Physica D*, 42(1-3):335–346, 1990.

[HCBS93] Henry Hexmoor, Guido Caicedo, Frank Bidwell, and Stuart C. Shapiro. Air battle simulation: An agent with "conscious" and "unconscious" layers. In *Proceedings of the University at Buffalo Graduate Conference on Computer Science*, 1993.

[HLCS93] Henry Hexmoor, Johan Lammens, Guido Caicedo, and Stuart C. Shapiro. Behavior based ai, cognitive processes, and emergent behaviors in autonomous agents. In *Proceedings of the International Conference of AI in Engineering (Toulouse, France, to appear)*, 1993.

[HLS92] Henry Hexmoor, Johan Lammens, and Stuart

Shapiro. An autonomous agent architecture for integrating perception and acting with grounded, embodied symbolic reasoning. Technical Report CS-92-21, Dept. of Computer Science, SUNY at Buffalo, New York (ftp.cs.buffalo.edu:/pub/tech-reports), 1992.

[HLS93] Henry Hexmoor, Johan Lammens, and Stuart Shapiro. Embodiment in glair: a grounded layered architecture with integrated reasoning for autonomous agents. In Douglas D. Dankel, editor, *Proceedings of the 6th Florida AI Research Symposium*, pages 325–329. Florida AI Research Society, 1993.

[HN92] Henry Hexmoor and Donald Nute. Methods for deciding *what to do next* and learning. Technical Report AI-1992-01, AI Programs, The University of Georgia, Athens, Georgia, 1992. Also available from SUNY at Buffalo, CS Department TR-92-23.

[JH55] Dorothea Jameson and Leo M. Hurvich. Some quantitative aspects of an opponent-colors theory: I. chromatic responses and spectral saturation. *JOSA*, 45(7):546–552, 1955.

[KM78] Paul Kay and Chad K. McDaniel. The linguistic significance of the meaning of basic color terms. *Language*, 54(3):610–646, 1978.

[LHS] Johan M. Lammens, Henry H. Hexmoor, and Stuart C. Shapiro. Of elephants and men. In *Proceedings of the NATO Advanced Studies Institute on the Biology and Technology of Intelligent Autonomous Agents (Trento, Italy, 3/1–12 1993)*. Springer Verlag (to appear. Preprint available as TR-93-13, Computer Science Department, SUNY at Buffalo (ftp.cs.buffalo.edu:/pub/tech-reports)), –.

[LMA91] Pat Langley, Kathleen McKusick, and John Allen. A design for the icarus architecture. In *ACM SIGART Bulletin*, pages 104–109. ACM publications, 1991.

[Mae90] Pattie Maes. Situated agents can have goals. In Pattie Maes, editor, *Designing Autonomous Agents*, chapter 4, pages 49–70. MIT Press, 1990.

[MRH86] J. L. McClelland, D. E. Rumelhart, and G. E. Hinton. The appeal of parallel distributed processing. In David Rumelhart, James McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing*, chapter 1, pages 3–44. MIT Press, Cambridge MA, 1986.

[Pol89] John Pollock. *How to Build a Person.* MIT Press, 1989.

[SR87] S. C. Shapiro and W. J. Rapaport. Sneps considered as a fully intensional propositional semantic network. In N. Cercone and G. McGalla, editors, *The knowledge frontier: essays in the representation of knowledge*, pages 262–315. Springer, New York, 1987.

[Wat89] Christopher Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge, UK, 1989.

[WS82] G. Wyszecki and W. S. Stiles. *Color Science.* Wiley, New York, 1982.

[Zad71] L. A. Zadeh. Quantitative fuzzy semantics. *Information Sciences*, 1971.