A Bottom-Up Model of Skill Learning

Ron Sun (rsun@cs.ua.edu)
Edward Merrill (emerrill@gp.as.ua.edu)
Todd Peterson (todd@cs.ua.edu)
The University of Alabama
Tuscaloosa, AL 35487

Abstract

We present a skill learning model CLARION. Different from existing models of high-level skill learning that use a top-down approach (that is, turning declarative knowledge into procedural knowledge), we adopt a bottom-up approach toward low-level skill learning, where procedural knowledge develops first and declarative knowledge develops later. CLARION is formed by integrating connectionist, reinforcement, and symbolic learning methods to perform on-line learning. We compare the model with human data in a minefield navigation task. A match between the model and human data is found in several respects.

Introduction

The acquisition and use of skill constitute a major portion of human activities. Skills vary in complexity and degree of cognitive involvement. They range from simple motor movements and other routine tasks in everyday activities to high-level intellectual skills. We study "lower-level" cognitive skills, which have not received sufficient research attention. One type of task that exemplifies what we call low-level cognitive skill is reactive sequential decision making (Sun et al 1996). It involves an agent selecting and performing a sequence of actions to accomplish an objective on the basis of moment-to-moment information (hence the term "reactive"). An example of this kind of task is the minefield navigation task developed at The Naval Research Lab (see Gordon et al. 1994). This kind of task setting appears to tap into real-world skills associated with decision making under conditions of time pressure and limited information. Thus, the results we obtain from human experiments will likely be transferable to real-world skill learning situations. Yet this kind of task is suitable for computational modeling given the recent development of machine learning techniques (Sun et al 1996, Watkins 1989).

The distinction between procedural knowledge and declarative knowledge has been made in many theories of learning and cognition (for example, Anderson 1982, 1993, Keil 1989, Damasio 1994, and Sun 1995). It is believed that both procedural and declarative knowledge are essential to cognitive agents in complex environments. Anderson (1982) originally proposed the distinction based on data from a variety of skill learning studies, ranging from arithmetic to geometric theorem proving, to account for changes resulting from extensive practice. Similar distinctions have been made by other researchers based on different sets of data, in the areas of skill learning, concept formation, and verbal informal reasoning (e.g., Fitts and Posner, 1967; Keil, 1989; Sun, 1995).

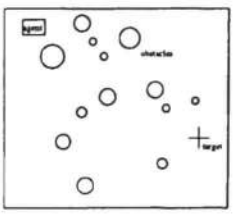

Figure 1: Navigating Through Mines

Most of the work in skill learning that makes the declarative/procedural distinction assumes a top-down approach; that is, learners first acquire a great deal of explicit declarative knowledge in a domain and then through practice, turn this knowledge into a procedural form ("proceduralization"), which leads to skilled performance. However, these models were not developed to account for skill learning in the absence of, or independent from, preexisting explicit domain knowledge. Several lines of research demonstrate that individuals can learn to perform complex skills without first obtaining a large amount of explicit declarative knowledge (e.g., Berry and Broadbent 1988, Stanley et al 1989, Lewicki et al 1992, Willingham et al 1989, Reber 1989, Karmiloff-Smith 1986, Schacter 1987, and Schraagen 1993). In research on *implicit learning*, Berry and Broadbent (1988), Willingham et al (1989), and Reber (1989) expressly demonstrate a *dissociation* between explicit knowledge and skilled performance in a variety of tasks including dynamic decision tasks (Berry and Broadbent 1988), artificial grammar learning tasks (Reber 1989), and serial reaction tasks (Willingham et al 1989). Berry and Broadbent (1988) argue that the psychological data in dynamic decision tasks are not consistent with exclusively top-down learning models, because subjects can learn to perform the task without being provided a priori declarative knowledge and without being able to verbalize the rules they used to perform the task. This indicates that procedural skills are not necessarily accompanied by explicit declarative knowledge, which would not be the case if top-down learning is the only way to acquire skill. Willingham et al (1989) similarly demonstrate that procedural knowledge

is not *always* preceded by declarative knowledge in human learning, and show that declarative and procedural learning are not necessarily correlated. There are even indications that explicit knowledge may arise from procedural skills in some circumstances (see Stanley et al 1989). Using a dynamic decision task, Stanley et al. (1989) found that the development of declarative knowledge paralleled but lagged behind the development of procedural knowledge.

Similar claims concerning the development of procedural knowledge prior to the development of declarative knowledge have surfaced in a number of research areas outside the skill learning literature and provided additional support for the bottom-up approach. *Implicit memory* research (e.g., Schacter 1987) demonstrates a dissociation between explicit and implicit knowledge/memories in that an individual's performance can improve by virtue of implicit "retrieval" from memory and the individual can be unaware of the process. This is not amenable to the exclusively top-down approach. *Instrumental conditioning* also reflects a learning process that differs from the top-down approach, because the process is typically non-verbal and involves the formation of action sequences without requiring explicit knowledge. It may be applied to simple organisms as well as humans (Gluck and Bower 1988). In *developmental psychology*, Karmiloff-Smith (1986) proposed the idea of "representational redescription". During development, low-level implicit representations are transformed into more abstract and explicit representations and thereby made more accessible. This process is not top-down either, but in the opposite direction.

The Model

The distinction between declarative and procedural knowledge leads naturally to "two-level" architectures (Sun 1995). We thereby developed the model CLARION, which stands for *Connectionist Learning with Adaptive Rule Induction ONline*. It embodies the distinction of declarative and procedural knowledge (or, conceptual and subconceptual knowledge), and it performs learning in a bottom-up direction. It consists of two main components: the top level encodes explicit declarative knowledge in the form of propositional rules, and the bottom level encodes implicit procedural knowledge in neural networks. In addition, there is an episodic memory, which stores recent experiences in the form of "input, output, result" (i.e., stimulus, response, and consequence).

A high-level pseudo-code algorithm that describes CLARION is as follows:

1. Observe the current state x.
2. Compute in the bottom level the Q-value of each of the possible actions (a_i's) associated with the perceptual state x: $Q(x, a_1), Q(x, a_2), ..., Q(x, a_n)$.
3. Find out all the possible actions (b_1, b_2, ..., b_m) at the top level, based on the the perceptual information x and other available information (which goes up from the bottom level) and the rules in place at the top level.
4. Choose an appropriate action a, considering the values of a_i's and b_j's (which are sent down from the top level).
5. Perform the action a, and observe the next state y and (possibly) the reinforcement r.
6. Update the bottom level in accordance with the *Q-Learning-Backpropagation* algorithm, based on the feedback information.
7. Update the top level using the *Rule-Extraction-Refinement* algorithm.
8. Go back to Step 1.

In the bottom level, a Q-value is an evaluation of the "quality" of an action in a given state: $Q(x, a)$ indicates how desirable action a is in state x. We can choose an action based on Q-values. To acquire the Q-values, supervised and/or reinforcement learning methods may be applied. A widely applicable option is the *Q-learning* algorithm (Watkins 1989), a reinforcement learning algorithm. In the algorithm, $Q(x, a)$ estimates the maximum discounted cumulative reinforcement that the agent will receive from the current state x on. The updating of $Q(x, a)$ is based on minimizing

$$r + \gamma e(y) - Q(x, a) \tag{1}$$

where γ is a discount factor, y is the new state resulting from action a in state x, and $e(y) = \max_a Q(y, a)$. Thus, the updating is based on the *temporal difference* in evaluating the current state and the action chosen: In the above formula, $Q(x, a)$ estimates, before action a is performed, the (discounted) cumulative reinforcement to be received if action a is performed, and $r + \gamma e(y)$ estimates the (discounted) cumulative reinforcement that the agent will receive, after action a is performed; so their difference (the temporal difference in evaluating an action) enables the learning of Q-values that approximate the (discounted) cumulative reinforcement. Using Q-learning allows sequential behavior to emerge in an agent. Through successive updates of the Q function, the agent can learn to take into account future steps in longer and longer sequences.

To implement Q functions, we chose to use a four-layered network (see Figure 2), in which the first three layers form a (either recurrent or feedforward) backpropagation network for computing Q-values and the fourth layer (with only one node) performs stochastic decision making. The output of the third layer (i.e., the output layer of the backpropagation network) indicates the Q-value of each action (represented by an individual node), and the node in the fourth layer determines probabilistically the action to be performed based on a Boltzmann distribution (i.e., Luce's choice axiom; Watkins 1989):

$$p(a|x) = \frac{e^{Q(x,a)/\alpha}}{\sum_i e^{Q(x,a_i)/\alpha}} \tag{2}$$

This learning process performs both structural credit assignment (with backpropagation), so that the agent knows which element in a state should be assigned credit/blame, as well as temporal credit assignment, so that the agent knows which action leads to success or failure. This learning process enables the development of procedural skills potentially solely based on the agent independently exploring a particular world on a continuous and on-going basis.

In the top level, declarative knowledge is captured in a simple propositional rule form. To facilitate correspondence with

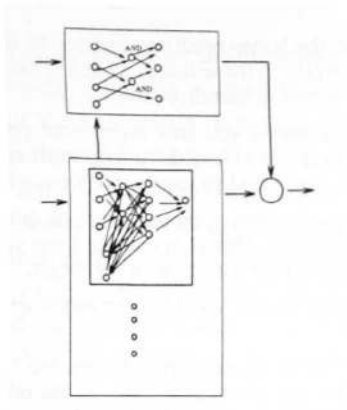

Figure 2: The implementation of CLARION.
The top level contains localist encoding of propositional rules. The bottom level contains connectionist networks for capturing procedural skills.

the bottom level and to encourage uniformity and integration (Clark and Karmiloff-Smith 1993), we chose to use a localist connectionist model for implementing these rules (e.g., Sun 1992). Basically, we translate the structure of a set of rules into that of a network. For each rule, a set of links are established, each of which connects a node representing an element in the condition of a rule to the node representing the conclusion of the rule. For more complex rule forms including predicate rules and variable binding, see Sun (1992).

To fully capture bottom-up learning processes, we devised a new algorithm for learning declarative knowledge (rules) using information in the bottom level (the *Rule-Extraction-Refinement* algorithm). The basic idea is as follows: if an action decided by the bottom level is "successful", then the agent extracts a rule (with its action corresponding to that selected by the bottom level and with its conditions corresponding to the current sensory state), and adds the rule to the top-level rule network. Then, in subsequent interactions with the world, the agent refines the extracted rule by considering the outcome of applying the rule: if the outcome is "successful", the agent may try to generalize ("expand") the condition of the rule to make it more universal; if the outcome is not successful, then the agent may specialize ("shrink") the condition of the rule.

Specifically, at each step, we compute an information gain measure that compares the qualities of two candidate rules. To do that, we examine the following information: (x, y, r, a), where x is the state before action a is performed, y is the new state after an action a is performed, and r is the reinforcement received after action a. Based on that, we update rule statistics: the positive match and the negative match counts for each rule condition and each of its minor variations (i.e., the rule condition plus/minus one possible value in one of the input dimensions) C, with regard to the action a performed; that is, $PM_a(C)$ (i.e., Positive Match, which equals the number of times that an input matches the condition C, action a is performed, and the result is positive) and $NM_a(C)$ (i.e., Negative Match, which equals the number of times that an input matches the condition C, action a is performed, and the result is negative). Here, positivity/negativity is determined by the following inequality: $\max_b Q(y, b) - Q(x, a) + r > threshold$, which indicates whether or not the action is reasonably good (Sun and Peterson 1998). Based on these statistics, we calculate the information gain measure; that is,

$$IG(A, B) = log_2 \frac{PM_a(A) + 1}{PM_a(A) + NM_a(A) + 2} - log_2 \frac{PM_a(B) + 1}{PM_a(B) + NM_a(B) + 2}$$

where A and B are two different conditions that lead to the same action a. The measure compares essentially the percentage of positive matches under different conditions A and B (with the Laplace estimator; Lavrac and Dzeroski 1994). If A can improve the percentage to a certain degree over B, then A is considered better than B. In the algorithm, if a rule is better compared with the match-all rule (i.e, the rule with the condition that matches all inputs), then the rule is considered "successful" (for the purpose of deciding on expansion or shrinking operations).

We decide on whether or not to construct a rule based on a simple success criterion which is fully determined by the current step (x, y, r, a):

- *Construction*: if $r + \gamma e(y) - Q(x, a) > threshold$, where a is the action performed in state x and y is the resulting new state [**that is, if the current step is successful**], and if there is no rule that covers this step in the top level, set up a rule $C \longrightarrow a$, where C specifies the values of all the input dimensions exactly as in x.

The criterion for applying the *expansion* and *shrinking* operators, on the other hand, is based on the afore-mentioned information gain measure. Expansion amounts to adding an additional value to one input dimension in the condition of a rule, so that the rule will have more opportunities of matching inputs, and shrinking amounts to removing one value from one input dimension in the condition of a rule, so that it will have less opportunities of matching inputs. Here are the detailed descriptions of these operators:

- *Expansion*: if $IG(C, all) > threshold1$ and $\max_{C'} IG(C', C) \geq 0$, where C is the current condition of an applicable rule, *all* refers to the match-all rule (with regard to the same action specified by the rule), and C' is a modified condition such that C' = C plus one value (i.e., C' has one more value in one of the input dimensions) [**that is, if the current rule is successful and an expanded condition is potentially better**], then set C'' = $argmax_{C'} IG(C', C)$ as the new (expanded) condition of the rule. Reset all the rule statistics. Any rule covered by the expanded rule will be placed in its children list. [1]

[1] The children list of a rule is created to keep aside and make inactive those rules that are more specific (thus fully covered) by the current rule. It is useful because if later on the rule is deleted or shrunk, some or all of those rules on its children list may be reactivated if they are no longer covered.

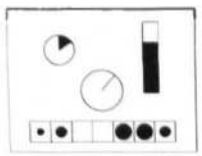

Figure 3: The Navigation Input

The display at the upper left corner is the fuel gauge; the vertical one at the upper right corner is the range gauge; the round one in the middle is the bearing gauge; the 7 sonar gauges are at the bottom.

- *Shrinking*: if $IG(C, all) < threshold2$ and $\max_{C'} IG(C', C) > 0$, where C is the current condition of an applicable rule, *all* refers to the match-all rule (with regard to the same action specified by the rule), and C' is a modified condition such that C' = C minus one value (i.e., C' has one less value in one of the input dimensions) [**that is, if the current rule is unsuccessful, but a shrunk condition is better**], then set $C'' = argmax_{C'} IG(C', C)$ as the new (shrunk) condition of the rule. Reset all the rule statistics. Restore those rules in the children list of the original rule that are not covered by the shrunk rule. If shrinking the condition makes it impossible for a rule to match any input state, delete the rule.

- *Deletion*: included in *Shrinking*.

Note that although the accumulation of statistics is gradual, the acquisition and revision of rules are one-shot and all-or-nothing, as opposed to the gradual changes in the bottom level.

In choosing an action to be performed at each step, we combine the corresponding values for each action from the two levels (at step 4 of the overall algorithm) by a weighted sum; that is, if the top level indicates that action a has an activation value v (which should be 0 or 1 as rules are binary) and the bottom level indicates that a has an activation value q (the Q-value), then the final outcome is $w_1 * v + w_2 * q$. Stochastic decision making with Boltzmann distribution (based on the weighted sums) is then performed to select an action out of all the possible actions (Willingham et al 1989). w_1 and w_2 are automatically determined through probability matching.

Experiments

In all of the human experiments, subjects were seated in front of a computer monitor that displayed an instrument panel containing several gauges that provided current information (see Figure 3). The following instruction was given to explain the setting:

> I. Imagine yourself navigating an underwater submarine that has to go through a minefield to reach a target location. The readings from the following instruments are available:
>
> (1) Sonar gauges show you how close the mines are to the submarine. This information is presented in 7 equal areas that range from 45 degrees to your left, to directly in front of you and then to 45 degrees to your right. Mines are detected by the sonars and the sonar readings in each of these directions are shown as circles in these boxes. A circle becomes larger as you approach mines in that direction.
>
> (2) A fuel gauge shows you how much time you have left before you run out fuels. Obviously, you must reach the target before you run out of fuel to successfully complete the task.
>
> (3) A bearing gauge shows you the direction of the target from your present direction; that is, the angle from your current direction of motion to the direction of the target.
>
> (4) A range gauge shows you how far your current location is from the target.
>
> II. At the beginning of each episode you are located on one side of the minefield and the target is on the other side of the minefield. You task is to navigate through the minefield to get to the target before you run out of fuel. An episode ends when: (a) you get to the goal (success); (b) you hit a mine (failure); (c) you run out of fuel (failure).

A random mine layout was generated for each episode. This setting was *stochastic* and *non-Markovian*. Because of the tight time limit, the subjects were forced to be reactive and use bottom-up learning. Five training conditions were used, in order to produce differences of performance resulting from differential emphases placed on the two levels respectively:

- The standard training condition. Subjects received five blocks of 20 episodes on each of five consecutive days (100 episodes per day). In each episode the minefield contained 60 mines. The subjects were allowed 200 steps.

- The verbalization conditions. They were identical to the standard condition except that subject were asked to step through replays of selected episodes and to verbalize what they were thinking during the episode. One group of subjects verbalized for five of the first 20 episodes and five of the last 20 episodes on the first, third, and fifth days, while another group verbalized on the fifth day only.

- The over-verbalization condition. Subjects were required to perform verbalization on 15 of the 25 episodes that they received during one session of training.

- The dual-task condition. Subjects performed the navigation task while concurrently performing a category decision task. (In the category decision task, subjects listened to a series of exemplars from five semantic categories at the rate of one every three seconds (on average). One category was designated the target category each day and subjects had to respond verbally when an exemplar of the category was presented.)

- The transfer conditions. Subjects were trained in 30 mine minefields until they reached the criterion of 80% success on two consecutive blocks. One group was trained under the single task condition, while the other under the dual task condition (as described earlier). Then they were both transfered to the 60 mine fields (without secondary tasks).

The rationale for designing these experiments was to manipulate training settings so as to allow differential emphases on the two levels in subjects, which serves to illustrate the

effects of the two levels in an indirect way, and thus verify the model indirectly (there is no way that we can verify the contributions of the two levels directly). For instance, with verbalization, subjects might be forced to be more explicit, and thus their top-level mechanisms might be more engaged and the performance enhanced to some extent (Stanley et al 1989, Willingham et al 1989). When subjects were forced to be completely explicit, their top-level mechanisms might be overly engaged and thus the bottom-level mechanisms might be hampered; thus the performance might be worsened (Reber 1989, Schooler et al 1993). When subjects were distracted by a secondary (explicit) task, their top-level mechanisms might be less available to the primary task (since attentional manipulation affects explicit processes more than implicit processes; Stadler 1995, Nissen and Bullemer 1987), which led to worsened performance.

CLARION was applied to the task in the same ways as human subjects. The effect of (regular) verbalization was posited to stem from heightened explication (rule learning) activities (Stanley et al 1989) and to a lesser extent, from rehearsing previous episodes. Thus for the model, we reduced the rule learning thresholds (to encourage more rule learning) and also used episodic memory replay (to capture rehearsal). To capture the effect of over-verbalization, we assumed that too much verbalization (e.g., verbalizing for more than half of the training episodes) reduced the rule learning thresholds even further. The effect of the dual task was conjectured to be hampering the top level. Thus in the model, the effect of the dual task was captured by significantly increasing the rule learning thresholds at the top level (which discouraged rule learning).

10 human subjects were compared to 10 model subjects (randomly selected) in each experiment. We obtained performance data for each subject separately. These were divided into blocks of 20 episodes each.

The effect of the dual task condition on learning. Success rates were averaged for each human or model subject. Comparing human and model performance with single vs. dual task training, 2x2 ANOVA (human vs. model x single vs. dual task) indicated a significant main effect for single vs. dual task ($p < .01$), but no interaction between groups and task types, indicating similar effects of the dual task condition on the learning of human and model subjects. See Figure 4.

The effect of the dual task condition on transfer. 2x2 ANOVA (human vs. model x single vs. dual task) revealed a significant main effect of single vs dual task ($p < .05$), and no interaction between groups and task types, again indicating similar effects of the dual task condition on the transfer of human and model subjects. See Figure 5.

The effect of verbalization. The effect was revealed by comparing performance of the two groups of verbalization subjects (one started verbalization on the first day and the other on the fifth day). The first four days were used to examine the effects of verbalization. We averaged success rates across each of these 4 days for each subject, and subjected

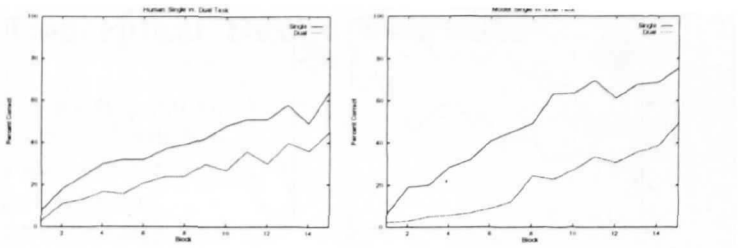

Figure 4: Single vs. Dual Task Training
The left panel contains averaged human data, and the right averaged model data.

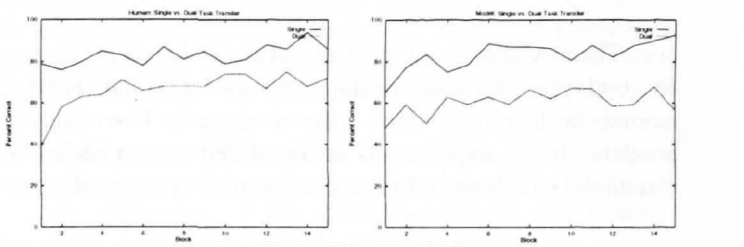

Figure 5: Single vs. Dual Task Transfer
The left panel contains averaged human data, and the right averaged model data.

the data to a 4 (days) x 2 (human vs. model) x 2 (verbalization vs. no verbalization) ANOVA. The analysis indicated that both human and model subjects exhibited a significant increase in performance due to verbalization ($p < .01$), but that the difference associated with verbalization for the two groups was not significant. See Figure 6.

The effect of over-verbalization. In the over-verbalization condition, virtually all subjects were performing at floor at the end of their 25 episodes of training. [2] CLARION captured this effect through the aforementioned reduction of the rule learning thresholds.

In addition, we compared the human and model subjects under the standard, the verbalization (starting the first day), and the dual-task condition. They were highly similar. The model data were within the standard error of the human data. Two corresponding sets of data in each condition were both best fit by power functions. A Pearson product moment correlation coefficient was calculated for each pair, which yielded high positive correlations (r ranged from .82 to .91), indicating a high degree of similarity between human and model subjects in how practice influenced human and model performance in each condition.

Concluding Remarks

In sum, we discussed a hybrid connectionist model CLARION as a demonstration of the approach of bottom-up skill learn-

[2] Overall, these subjects achieved a 10% success rate, whereas the subjects in the regular verbalization condition achieved a success rate of 33%. If we eliminate the one subject who performed at 60% in the over-verbalization condition, the remaining subjects achieved a success rate of approximately 3%.

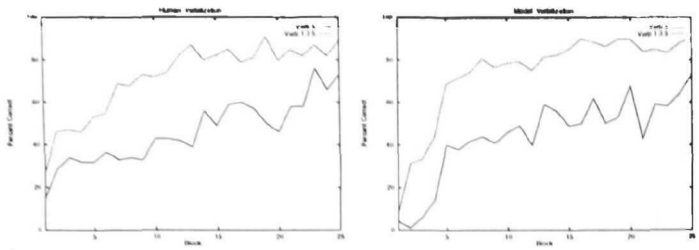

Figure 6: Verbalization vs. No Verbalization
The left panel contains averaged human data, and the right averaged model data.

ing. The model essentially consisted of two levels for capturing both procedural and declarative knowledge and enabling bottom-up learning, which differed markedly from existing models. Initial experiments demonstrated some matches of the model with human data across a number of manipulations.

Acknowledgements

This work is supported in part by Office of Naval Research grant N00014-95-1-0440. The simulator was provided by NRL.

References

J. R. Anderson, (1982). Acquisition of cognitive skill. *Psychological Review*. Vol.89, pp.369-406.

J. R. Anderson, (1993). *Rules of the Mind*. Lawrence Erlbaum Associates. Hillsdale, NJ.

D. Berry and D. Broadbent, (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology*. 79, 251-272.

A. Clark and A. Karmiloff-Smith, (1993). The cognizer's innards: a psychological and philosophical perspective on the development of thought. *Mind and Language*. 8 (4), 487-519.

A. Cleeremans and J. McClelland, (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*. 120. 235-253.

A. Damasio, (1994). *Decartes' Error*. Grosset/Putnam, NY.

D. Gordon, et al. (1994). *User's Guide to the Navigation and Collision Avoidance Task*. Naval Research Lab. DC.

H. Dreyfus and S. Dreyfus, (1987). *Mind Over Machine: The Power of Human Intuition*, The Free Press, New York, NY.

P. Fitts and M. Posner, (1967). *Human Performance*. Brooks/Cole, Monterey, CA.

M. Gluck and G. Bower, (1988). From conditioning to category learning. *Journal of Experimental Psychology: General*. 117 (3), 227-247.

W. James, (1890). *The Principles of Psychology*. Dover, NY.

A. Karmiloff-Smith, (1986). From meta-processes to conscious access. *Cognition*. 23. 95-147.

F. Keil, (1989). *Concepts, Kinds, and Cognitive Development*. MIT Press. Cambridge, MA.

N. Lavrac and S. Dzeroski, (1994). *Inductive Logic Programming*. Ellis Horword, New York.

P. Lewicki, et al. (1992). Nonconscious acquisition of information. *American Psychologist*. 47, 796-801.

M. Nissen and P. Bullemer, (1987). Attentional requirements of learning: evidence from performance measures. *Cognitive Psychology*. 19, 1-32.

A. Reber, (1989). Implicit learning and tacit knowledge. *Journal of Exp Psychology: General*. 118 (3), 219-235.

D. Schacter, (1987). Implicit memory: History and current status. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13, 501-518.

J. Schooler, S. Ohlsson, and K. Brooks, (1993). Thoughts beyond words: when language overshadows insight. *Journal of Experimental Psychology: General*. 122 (2). 166-183.

J. Schraagen, (1993). How experts solve a novel problem in experimental design. *Cognitive Science*. 17, 285-309.

P. Smolensky, (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11(1):1–74.

M. Stadler, (1995). Role of attention in implciit learning. *Journal of Experimental Psychology: Learning, Memory and Cognition*. 15, 1061-1069.

W. Stanley, et al (1989). Insight without awareness *Quarterly Journal of Experimental Psychology*. 41A (3), 553-577.

R. Sun, (1992). On Variable Binding in Connectionist Networks, *Connection Science*, Vol.4, No.2, pp.93-124.

R. Sun, (1995). Robust reasoning. *Artificial Intelligence*. 75, 2. 241-296.

R. Sun, T. Peterson, and E. Merrill, (1996). Bottom-up skill learning. *Proc.of 18th Cognitive Science Society Conference*,

R. Sun and T. Peterson, (1998). Some experiments with a hybrid model for learning sequential decision making. *Information Sciences*. in press.

R. Sutton, (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proc.of Seventh International Conference on Machine Learning*. Morgan Kaufmann. San Mateo, CA.

T. Tesauro, (1992). Practical issues in temporal difference learning. *Machine Learning*. Vol.8, 257-277.

C. Watkins, (1989). *Learning with Delayed Rewards*. Ph.D Thesis, Cambridge University, Cambridge, UK.

D. Willingham, M. Nissen, and P. Bullemer, (1989). On the development of procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 15, 1047-1060.