Bottom-up Skill Learning in Reactive Sequential Decision Tasks

Ron Sun and **Todd Peterson** and **Edward Merrill**
The University of Alabama
Tuscaloosa, AL 35487
rsun@cs.ua.edu

Abstract

This paper introduces a hybrid model that unifies connectionist, symbolic, and reinforcement learning into an integrated architecture for bottom-up skill learning in reactive sequential decision tasks. The model is designed for an agent to learn continuously from on-going experience in the world, without the use of preconceived concepts and knowledge. Both procedural skills and high-level knowledge are acquired through an agent's experience interacting with the world. Computational experiments with the model in two domains are reported.

Introduction

Skill learning (or skill acquisition) is an important area of cognitive science, as skilled performance (and its acquisition) constitutes the majority of human activities. Such skills range from simple motor movements and routine coping in everyday activities all the way to complex intellectual skills such as writing or proving mathematical theorems. There is a hierarchy of skills of varying complexities and cognitive involvement. Most widely studied in cognitive science is cognitive skill acquisition (VanLehn 1995), that is, the abilities to solve problems in more or less intellectual tasks, such as (just to mention a few) arithmetic, elementary geometry, LISP programming, and simulated airtraffic control (e.g., Anderson 1982, 1993, VanLehn 1995, Ackerman 1988). Most of the work assumes a top-down approach; that is, they assume that subjects first acquire a great deal of knowledge in a domain and then practice changes this explicit knowledge into a more usable form, which leads to skilled performance. The explicit knowledge acquired before practice is declarative knowledge while the knowledge directly used in skilled performance is procedural knowledge. It is commonly believed that skills are the result of "proceduralization" of declarative knowledge.

However, there is a substantial literature of work that demonstrates that the opposite may also be true: subjects can learn skilled performance without being provided explicit knowledge prior to practice, such as Berry and Broadbent (1984), Stanley et al (1989), Willingham et al (1992), and Reber (1989). Among them, Berry and Broadbent (1984) and Stanley et al (1989) expressly demonstrate the *dissociation* between prior knowledge and skilled performance, in a variety of tasks. Explicit knowledge is not equivalent to but can arise out of skills.

Reactive sequential decision tasks (Sun and Peterson 1995) is a suitable domain for studying such *bottom-up* skill learning. They generally involve selecting and performing a sequence of actions, in order to accomplish an objective, mostly on the basis of moment-to-moment perceptual information. In such tasks, while skills emerge from repeated practice, declarative knowledge is also formed, on the basis of acquired skilled performance. So the process is the opposite of the commonly assumed top-down approach.

A general specification is as follows: there is an agent that can select, from a finite set of actions, a particular action to perform at each time step. The selection decision is (mainly) based on the current state of the world, presented to the agent through sensory input. The world changes either autonomously or as a result of some action by an agent. Thus, over time, the world is presented to an agent as a sequence of states. At certain points in a sequence, the agent may receive *payoffs* or *reinforcements*. Thus, the agent may need to perform temporal and structural *credit assignment*, to attribute the *payoffs/reinforcements* to various actions at various points in time (that is, the temporal credit assignment problem), in accordance to various aspects of a state (that is, the structural credit assignment problem).

While performing this kind of task, the agent is often under severe time pressure. Often a decision has to be made in a fraction of a second; therefore it cannot do much "information processing", and falls outside of Allen Newell's "rational band". The decision making and learning in the agent thus cannot be too time-consuming. As in humans, the agent may also be severely limited in other resources, such as memory so that memorizing all the previous episodes is considered impossible. The perceptual ability of an agent may also be extremely limited so that only very local information is available. Learning in such a domain is an experiential, trial-and-error process; the agent develops knowledge *tentatively* on an on-going basis, since it cannot wait until the end of an episode. Learning is thus *concurrent* or on-line (Nosofsky et al 1994).

Hybrid Models

In such tasks, with bottom-up learning and without prior knowledge, how can an agent develop a set of coping skills that are highly specific (geared towards particular situations) and thus highly efficient but, at the same time, acquire sufficiently general knowledge that can be readily applied to a variety of different situations? In the current context, one way to learn is through trial-and-error: repeated practice gradually gives rise to a set of procedural skills that deal specifically with the practiced situations and their minor variations. However, such skills may not be transferable to truly novel situations, since they are so embedded in specific contexts and tangled

together. The agent needs both procedural and declarative knowledge, or both subconceptual and conceptual knowledge. It is assumed that a balance of the two is essential to the development of complex cognitive agents. Generic declarative knowledge, which can emerge from procedural skills, has the following three advantages: (1) It helps to guide the exploration of novel situations, and reduces the time (i.e., the number of trials) necessary to develop specific skills in new situations. In other words, it helps the transfer of learned skill (as shown through psychological data by Willingham et al 1989). (2) Generic knowledge can help to speed up learning. If properly used, generic knowledge that is extracted on-line during learning can help to facilitate the very learning process itself. (3) Generic knowledge can also help in communicating learned knowledge and skills to other agents.

Two-level hybrid models seem to provide the needed computational framework for representing both types of knowledge (Sun and Bookman 1994). Based on the ideas proposed in Sun (1994, 1995), we developed CLARION. See Figure 1. The bottom level contains specific procedural knowledge (Anderson 1982). The top level contains generic declarative knowledge. An overall pseudo-code algorithm is as follows:

1. Observe the current state x (in a proper representation).
2. Compute in the bottom level the Q-values of x associated with each of all the possible actions: $Q(x, a_1)$, $Q(x, a_2)$, ..., $Q(x, a_n)$.
3. Find out all the possible actions (b_1, b_2, ..., b_m) at the top level, based on the input x and the rules in place.
4. Compare the values of a_i's with those of b_j's, and choose an appropriate action b
5. Perform the action b, and observe the next state y and (possibly) the reinforcement r.
6. Update Q-values in accordance with the Q-learning algorithm
7. Update the rule network in the top level using the RULE-EXTRACTION-GENERALIZATION-REVISION.
8. Go back to Step 1.

In terms of representation in the bottom level, we prefer a subsymbolic distributed representation, such as that provided by a backpropagation network. (Existing evidence indicates that the difference between the two levels lies primarily in their representations; see Reber 1989.) This is because of the implicit nature of procedural skills: there is generally a lack of conceptual-level thinking in performing procedural skills; as a consequence, details of such skills are in general inaccessible (Anderson 1982, Ackerman 1988). A distributed representation naturally captures this property of procedural skills (Sun 1994), with representational units that are capable of accomplishing tasks but are in general uninterpretable and subsymbolic. (Otherwise, a symbolic representation may be used, but then we will have to artificially assume that these representations are not accessible, while some other similar representations are accessible — the distinction is arbitrary and not intrinsic to the media of representations; see Anderson 1993 and also Rosenbloom et al. 1993 regarding accessability of symbolic structures).

In terms of learning, we use reinforcement learning (the temporal difference method). 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 (which consists of some sketchy sensory input). To acquire the Q-values, we use *Q-learning* (Watkins 1989). In the algorithm, $Q(x, a)$ estimates the maximum discounted cumulative reinforcement that the agent will receive from the current state x on: $\max(\sum_{i=0}^{\infty} \gamma^i r_i)$, where γ is a discount factor that favors reinforcement received sooner relative to that received later, and r_i is the reinforcement received at step i (which may be 0). The updating of $Q(x, a)$ is based on minimizing $r + \gamma e(y) - Q(x, a)$, where γ is a discount factor 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. 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 combine Q-learning with connectionist representation, we use a four-layered network (see Figure 1) in which the first three layers form a 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 (Watkins 1989): $p(a|x) = \frac{e^{1/\alpha Q(x,a)}}{\sum_i e^{1/\alpha Q(x,a_i)}}$, where α controls the degree of randomness (temperature) of the decision-making process. The training of the network is based on minimizing the temporal difference as specified before.

Declarative knowledge is handled differently. For declarative knowledge, we prefer a symbolic or localist representation, in which each unit has a clear conceptual meaning or interpretation. This allows declarative knowledge to be highly accessible and inferences to be performed explicitly at a conceptual level (Smolensky 1988, Sun 1994). Because declarative knowledge needed in reactive sequential decision tasks is relatively simple, we will focus on propositional rules. We use a localist connectionist model (see Figure 1) for representing these rules to facilitate correspondence with the bottom level and to encourage uniformity and integration. Basically, we connect the nodes representing conditions of a rule to the node representing the conclusion. However, we need to wire up rules involving conjunctive conditions. For details, see Sun (1992).

Because of the dynamic nature of reactive sequential decision tasks, we need to be able to dynamically acquire a rule representation and to modify it in subsequent encounters if necessary. We thus need a simple and efficient way. We can make use of the bottom level which is trained with reinforcement learning to perform specific procedural skills by extracting information from the network (Towell and Shavlik 1993) and thereby forming and modifying explicit rules. The basic idea for rule learning is as follows: if some action decided by the bottom level is successful the agent extracts a rule that corresponds to the action selected by the bottom level and adds the rule to the top level. Then, in subsequent interactions with the world, the agent tries to verify the extracted rule, by considering the outcome of applying the rule: if the outcome is not successful, then the rule should be made more specific and exclusive of the current case; if the outcome is successful, the agent may try to generalize the rule to make it

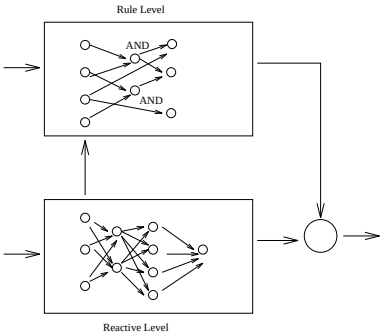

Figure 1: The CLARION Architecture

more universal (Mitchell 1982).

Specifically, three different criteria can be used for extracting rule from the bottom level: (1) direct reinforcement received at a step, (2) temporal difference (as used in updating Q-values), and (3) maximum Q-values in a state. The first criterion is an indication of whether or not an action taken in a given state is *directly* beneficial, but it fails to take into account sequences of actions. The second criterion indicates if further improvement in a Q-value is possible. The third criterion concerns whether the Q-value of a state and an action is close enough to the maximum Q-value in that state, indicating whether that action is close to being optimal in that state. (See Sun and Peterson 1995 for an analysis of these criteria.) We adopt a three-phase approach here, with three criteria being successively applied in different phases. At each step, after an action is selected and performed in a state, a new state is entered and reinforcement is received. Then, one of the three measures above that is applicable to the current phase is compared to a threshold to determine if a rule should be extracted. If so, a rule is formed that relates the state to the action, and the rule is then wired up in the top-level rule network.

After a rule is extracted, generalization and revision operations are used to tune the rule:

- *Expansion:* the value range of a condition is expanded by one interval, when a rule is successfully applied according to the criterion in the current phase.
- *Shrinking*: when a rule leads to unsuccessful results as judged by the criterion in the current phase, we reduce the value ranges of some or all conditions (cf. Michalski et al 1986).
- *Deletion*: remove a rule from the rule network when a counter example to the original case from which the rule was extracted is encountered, according to the current-phase criterion
- *Merge*: when the conditions of two rules are close enough, the two rules may be combined so that a more general rule can be produced.

The necessity of having a two-level architecture can be summed up as follows: (1) Without the bottom level, the agent will not be able to represent procedural skills sufficiently. Such skills may involve graded, uncertain, and inconsistent knowledge and autonomous stochastic exploration (with numeric calculation and probabilistic firing). (2) Without learning in the bottom level, the agent will not be able to learn from experience, and therefore will not be able to dynamically acquire either procedural skill in the bottom level,

Figure 2: The initial maze
The starting position is marked by 'S' in which the agent faces upward to the upper wall. The goal is marked by 'G'.

or rules in the top level (as in the current model). The bottom level also captures the gradual learning of skills, different from one-shot rule learning. (3) Without the top level, the agent will not be able to represent generic, easily accessible, and crisp knowledge and explicitly access and communicate that knowledge to other agents. When novel situations are encountered and/or when precision, crispness, consistency, and certainty are needed, declarative knowledge is preferred. Explicit access and explanation is also important in facilitating cooperation among agents. (4) Without rule learning, the agent will not be able to acquire *dynamically* and quickly crisp conceptual knowledge for the top level, and therefore has to resort to mostly pre-wired and/or externally given knowledge in the top level.

We try two different methods of combining outcomes from the two levels. One is the percentage method, and the other is the stochastic method. In the percentage method, in (randomly chosen) p percent of the steps, we use the outcome from the rule level, if there is at least one rule indicating an action in the current state; otherwise, we use the outcome of the bottom level (which is always available). In the stochastic method, we combine the corresponding values for each action from the two levels by a weighted sum; that is, if the top level indicates that action a has an activation value v and the bottom level indicates that a has a value q (the Q-value for a), then the weighted sum is $w_1 * v + w_2 * q$. Based on these weighted sums, stochastic decision making (with Boltzmann distribution) is then performed to select an action. The parameters, w_1, w_2, and p, are to be varied.

Experiments

Experiments with Mazes

We carried out some computational experiments in reactive sequential decision domains to show the advantage of the model in learning and transfer as hypothesized earlier. In a simple maze as in Figure 2, the agent has rudimentary sensory inputs regarding its immediate left, front and right side, indicating whether there is a wall, an opening, or the goal; the agent can move forward, turn to the left, or turn to the right. It has no information regarding its location except the simple sensory input described above. Each episode starts with an agent at a fixed starting location and ends when the agent reaches the goal (Figure 2). The reward for an agent reaching the goal is 1, and the punishment for hitting a wall is -0.1.

We first choose (optimize) the structures and parameters of backpropagation and Q-learning through trial-and-error: 8 hidden units are used, the learning rate is 0.1, the momentum parameter is 0.7, network weights are randomly initialized between -0.01 and 0.01; the Q-value discount rate is 0.9, the

	Moves	Rules
Q-learning	15348.48	n/a
Perc.60	4994.52	7.78
Perc.80	5840.14	7.28
Perc.60.gen	5164.36	8.50
Perc.80.gen	5040.84	9.12
Stoc.15.	4602.88	6.62
Stoc.20	4712.70	6.30
Stoc.15.gen	6539.04	6.82
Stoc.20.gen	5574.24	8.14

Figure 3: A Comparison of Learning Speeds

Moves indicate the total numbers of moves during training (averaged over 50 trials). *Rules* indicate the average numbers of rules at the end of training.

randomness parameter for stochastic decision making is set at 0.1. (Note that although these parameters make some differences, performance is not overly sensitive to small variations of their settings.) The lengths of phase 1, 2 and 3 are 3, 20, and 37 episodes, respectively.

Figure 3 shows the differences in learning speed, where learning speed is measured by the total number of moves in the first 60 episodes. *Perc.x* refers to the versions using the percentage combination with rules being applied $p = x\%$ of the times. *Stoc.y* refers to the versions using the stochastic combination with rules being weighted at $y\%$. The symbol *gen* indicates that generalization/revision operations (i.e., *expansion*, *shrinking*, etc.) on the extracted rules are performed; otherwise, none of these operations is performed. We recorded the results averaged over 50 trials with different random seeds. It is clear from the figure that, when rules are used frequently (e.g., with Perc.80 or Stoc.20), CLARION learns faster than pure Q-learning by large margins. A t test showed the differences were significant with over 99% confidence ($p < 0.01$). The data also indicates that generalization per se did not lead to faster learning.

In Figure 4, we show the average number (averaged over 50 trials) of steps needed to reach the target in one episode, after 60 episodes of training, for different models. The numbers are shown in the *Moves* column. The different versions of CLARION again outperform pure Q-learning by large margins. T tests showed over 99% confidence ($p < 0.01$). Also reported are the average numbers of steps in one episode, after the training, using only the top level (marked as *R-moves*) or using only the bottom level (marked as *Q-moves*). There is a *synergy* between the two levels: Comparing the three values horizontally on each line, the whole CLARION system always performs better than the top level alone or the bottom level alone.

We applied our trained models (after the training of 60 episodes) to a new and larger maze as shown in Figure 5 to access transfer. Transfer occurs because of the similarity of the two mazes. In Figure 6, as indicated by the *Moves* column, the different versions of CLARION transfer much better than Q-learning alone in terms of number of steps to reach the goal in one episode. Furthermore, by comparing the corresponding *Moves*, *Q-moves*, and *R-moves* on each line, we see that often learned rules alone perform better in transfer than the Q-learning network at the bottom level, as well as than the whole CLARION model. The superiority of R-moves

Maze 1

	Moves	Q-Moves	R-Moves
Q-learning	149.00	149.00	n/a
Perc.60	29.76	72.46	94.98
Perc.80	10.78	36.22	13.48
Perc.60.gen	42.06	118.24	189.18
Perc.80.gen	22.02	55.14	106.58
Stoc.15	28.42	102.70	44.74
Stoc.20	20.60	81.80	30.54
Stoc.15.gen	53.90	87.18	108.20
Stoc.20.gen	36.26	67.18	64.66

Figure 4: Trained Performance

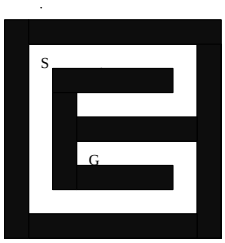

Figure 5: The second maze

in comparison with Q-moves demonstrates that it is rule induction that facilitates transfer to new and more complicated environments.

We also applied the trained model to an even larger maze as in Figure 7. The result is similar and the same points can be made in this case.

Experiments with Navigation

To further demonstrate CLARION, we tested it on a more complex task: the simulated navigation task. The agent has to navigate an underwater vessel to go through a minefield to reach a target location. The agent receives information only from a number of instruments. The sonar gauge shows how close the mines are in 7 equal areas that range from 45 degrees to the left of the agent to 45 degrees to the right. The fuel gauge shows the agent how much time is left before fuel runs out. The bearing gauge shows the direction of the target from the present direction of the agent. The range gauge shows how far the target is from the current location. Using such limited information, the agent decides on (1) how to turn and (2) how fast to move. The agent, within an allotted time period, can either (a) reach the target (which is a success), (b) hit a mine (a failure), or (c) run out of fuel (a failure again).

Maze 2

	Moves	Q-Moves	R-Moves
Q-learning	1681.48	1681.48	n/a
Perc.60	770.72	1782.16	559.96
Perc.80	492.14	1289.78	233.56
Perc.60.gen	766.38	2049.40	1030.66
Perc.80.gen	415.52	1581.62	722.48
Stoc.15	850.70	1481.34	405.94
Stoc.20	498.40	1586.88	392.08
Stoc.15.gen	703.80	1690.32	981.94
Stoc.20.gen	760.70	2028.24	956.50

Figure 6: Transfer to Maze 2

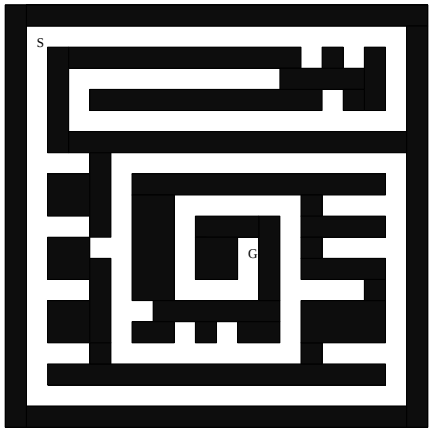

Figure 7: The third maze

	Successful Episodes
Q-learning	38.1
Stoc.10	278.9
Stoc.20	301.5
Stoc.10.gen	301.3
Stoc.20.gen	254.2

Figure 8: Learning
The number of successful episodes during training is included for each case.

In this experiment, each time the minefield is generated anew in a random layout, but it always contains the same number of mines, which in this case is 10. The time alloted to the agent for each episode is 200 steps. Figure 8 shows learning differences, where learning is measured by the total number of successful episodes out of a total 500 training episodes. CLARION again outperforms Q-learning alone.

Discussions

Most of the existing cognitively-motivated models for skill learning that contain both declarative and procedural knowledge explore mainly top-down learning, such as Anderson (1982, 1993), Gelfand et al (1989), and Schneider and Oliver (1991). CLARION explores bottom-up learning, to demonstrate how conceptual/symbolic knowledge can emerge through interacting with the world in the same way as subconceptual procedural knowledge does, and the performance advantage of such emergence.

In addition, while some other hybrid connectionist models try to implement all types of knowledge, symbolic and non-symbolic, in one kind of network or another (Miikkulainen and Dyer 1991, Barnden 1988, Sun 1992), CLARION takes a different tack and attempts to develop a principled dichotomy of the conceptual vs. the subconceptual in hybrid architectures. CLARION attempts to explore their synergy so that it learns faster and transfers better.

Some existing hybrid models do not, or cannot, perform learning (Sun 1992), while others perform learning in a batch fashion (e.g., Miikkulainen and Dyer 1991) and are thus cognitively implausible in this aspect. In contrast to these hybrid models, CLARION is capable of incremental, on-line (concurrent) learning, and *integrative* learning, that is, developing connectionist and symbolic representation along side of each other.

Acknowledgements

This work is supported in part by Office of Naval Research grant N00014-95-1-0440. We wish to thank Susan Chipman, Helen Gigley, Dave Waltz, Devika Subramanian, Diana Gordon, Jim Ballas and Alan Schultz for various helps.

References

P. Ackerman, (1988). Determinants of individual differences during skill acquisition: cognitive abilities and information processing. *Journal of Experimental Psychology: General*. 1117 (3), 288-318.

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

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

J. Barnden, (1988). The right of free association. *Proc.10th Conference of Cognitive Science Society*, 503-509, Lawrence Erlbaum Associates, Hillsdale, NJ.

D. Berry and D. Broadbent, (1984). On the relationship between task performance and associated verbalizable knowledge. *Quarterly Journal of Experimental Psychology*. 36A, 209-231.

J. Gelfand, D. Handelman and S. Lane, (1989). Integrating Knowledge-based Systems and Neural Networks for Robotic Skill Acquisition, *Proc.IJCAI*, pp.193-198. Morgan Kaufmann, San Mateo, CA.

J. Kruschke, (1992). ALCOVE: an examples-based connectionist model of category learning. *Psychological Review*. 99, 22-44.

R. Maclin and J. Shavlik, (1994). Incorporating advice into agents that learn from reinforcements. *Proc.of AAAI-94*. Morgan Kaufmann, San Meteo, CA.

R. Michalski et al., (1986). The multi-purpose incremental learning system AQ15. *Proc.of AAAI-86*. 1041-1045. Morgan Kaufmann. San Mateo, CA.

R. Miikkulainen & M. Dyer, (1991). Natural language processing with modular PDP networks and distributed lexicons. *Cognitive Science*. 15(3). pp.343-399.

T. Mitchell, (1982). Generalization as search. *Artificial Intelligence*, 18, 203-226.

R. Nosofsky, T. Palmeri, and S. McKinley, (1994). Rule-plus-exception model of classification learning. *Psychological Review*. 101 (1), 53-79.

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

P. Rosenbloom, J. Laird, and A. Newell, (1993). *The SOAR papers*. MIT Press.

W. Schneider and W. Oliver (1991), An instructable connectionist/control architecture. In: K. VanLehn (ed.), *Architectures for Intelligence*, Erlbaum, Hillsdale, NJ.

R. Shriffrin and W. Schneider, (1977). Controlled and automatic human information processing II. *Psychological Review*. 84. 127-190.

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

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, (1994). *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. John Wiley and Sons, New York, NY.

R. Sun, (1995). Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence*. 75, 2. 241-296.

R. Sun and L. Bookman, (eds.) (1994). *Computational Architectures Integrating Neural and Symbolic Processes*. Kluwer Academic Publishers. Norwell, MA.

R. Sun and T. Peterson, (1995). A hybrid learning model of reactive sequential decision making. *The Working Notes of The IJCAI Workshop on Connectionist-Symbolic Integration*.

G. Towell and J. Shavlik, (1993). Extracting Refined Rules from Knowledge-Based Neural Networks, *Machine Learning*.

K. VanLehn, (1995). Cognitive skill acquisition. *Annual review of Psychology*, Vol.47, J. Spence, J. Darly, and D. Foss (eds.) Annual Reviews, Palo Alto, CA.

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

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