From Implicit Skills to Explicit Knowledge: A Bottom-Up Model of Skill Learning

Ron Sun
CECS, University of Missouri-Columbia, Columbia, MO 65211
rsun@cecs.missouri.edu
573-884-7662

Edward Merrill
Todd Peterson
University of Alabama, Tuscaloosa, AL 35487

December 8, 1999

To appear in: Cognitive Science

Abstract

This paper presents a skill learning model Clarion. Different from existing models of mostly high-level skill learning that use a top-down approach (that is, turning declarative knowledge into procedural knowledge through practice), we adopt a bottom-up approach toward low-level skill learning, where procedural knowledge develops first and declarative knowledge develops later. Our model is formed by integrating connectionist, reinforcement, and symbolic learning methods to perform on-line reactive learning. It adopts a two-level dual-representation framework (Sun 1995), with a combination of localist and distributed representation. We compare the model with human data in a minefield navigation task, demonstrating some match between the model and human data in several respects.

1 Introduction

The acquisition and use of skill constitute a major portion of human activities. Naturally the study of skill learning is a major focus of research in cognitive science. The most widely studied in cognitive science is cognitive skill acquisition (e.g., VanLehn 1995), which refers to learning to solve problems in intellectually oriented tasks such as arithmetic, puzzle solving, elementary geometry, and LISP programming (Anderson 1982, 1993, VanLehn 1995, Rosenbloom et al 1993). However, 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. Therefore, besides studying highly intellectual tasks, it is also important to study "lower"-level cognitive skills, which have not received sufficient research attention in cognitive modeling. Our goal in this paper is to develop a computational model of low-level cognitive skill learning, which differs from existing work markedly.

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). In the task, an agent is required to navigate through a minefield to reach a target (see Figure 1). The agent is provided with only limited sensory information — short-range rudimentary sonar readings that allow it to see a small section of space in front of it. The agent has to reach the target in a limited amount of time (see section 3.1 for details). 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 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 (Watkins 1989, Barto et al 1996, Sun et al 1996, Sun and Peterson 1997, 1998, 1998b).

Another motivation for this work is to model learning that does not require a large amount of a priori knowledge (in the form of instructions or examples). Most existing models of skill learning are "top-down" (Sun et al 1996, Sun and Peterson 1997, 1998): they generally assume that individuals learn generic, verbal, declarative knowledge first and then through practice, turn such knowledge into specific, usable procedural skill (Anderson 1983, 1993, Ackerman 1988). However, when individuals are not provided a sufficient amount of a priori knowledge relevant to a task, learning may proceed differently. We believe that it is likely that some skills develop prior to the learning of declarative knowledge, with explicit declarative knowledge being constructed only after the skill is at least partially developed. We believe that we can capture this type of "bottom-up" learning in low-level skill domains.

In the remainder of this paper, we will first develop a skill learning model and its detailed im-

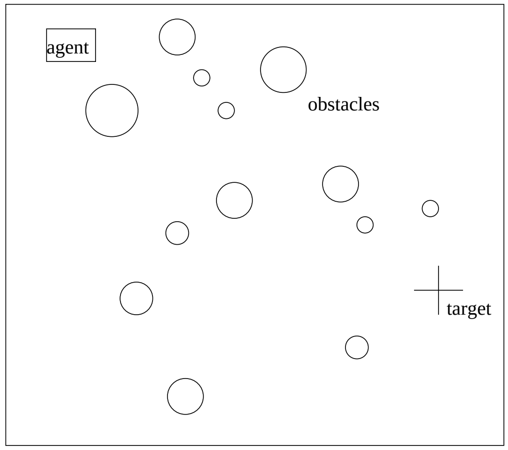

Figure 1: Navigating Through Mines

plementation and discuss its rationale (section 2). We will then present some comparisons between human experiments and computational simulations using the model (section 3). Discussions will follow, which further elaborate the model and compare it to related work (section 4). Note that this work develops a broad model that is meant to be generic (but can be tailored to specific tasks). Generality is emphasized rather than extremely fine-grained modeling. Our results show some support for the model but do not uniquely support the model.

2 Developing Models

2.1 Background

To justify the model development, we will discuss (1) the distinction between procedural and declarative knowledge, (2) the on-line nature of skill learning, and (3) the bottom-up learning that proceeds from procedural to declarative knowledge.

Procedural vs. Declarative Knowledge. The distinction between procedural knowledge and declarative knowledge has been made in many theories of learning and cognition (for example, Anderson 1976, 1983, 1993, Keil 1989, Damasio et al. 1994, and Sun 1994). It is believed that both procedural and declarative knowledge are essential to cognitive agents in complex environments. Anderson (1983) 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. For Anderson, the initial stage of skill development is characterized by the acquisition of declarative knowledge (i.e., explicit knowledge of how to perform the task). During this stage, the learner must attend to this declarative knowledge in order to successfully perform the task. Through practice, a set of specific procedures develop that allows aspects of the skill to be performed without using declarative knowledge. When the skill is proceduralized, it can be performed with almost no access to declarative knowledge and often even without concurrent conscious awareness of the specific details involved. 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, 1994).

Several other distinctions made by researchers capture a similar difference between different types of processing. For example, Smolensky (1988) proposed a distinction between conceptual (publicly accessible) and subconceptual (inaccessible) processing. In this framework, the use of declarative knowledge is based on conceptual processing and the use of procedural knowledge is based on subconceptual processing (and thus inaccessible). The inaccessibility of procedural knowledge is accepted by most researchers and embodied in most computational models that capture procedural skills, including Anderson (1983, 1993) and Rosenbloom, Laird, and Newell (1993). Dreyfus and Dreyfus (1987) proposed the distinction of analytical and intuitive thinking, and believed that the transition from the former to the latter is essential to the development of complex cognitive skills (on the basis of phenomenological analyses of chess playing at different stages of learning chess). This transition is very similar to the declarative-to-procedural transition as advocated by Anderson (1983, 1993), although they are not exactly identical. In addition, the distinction between conscious and unconscious processing (cf. James 1890, Reber 1989, Lewicki et al 1992) fits this framework in that declarative knowledge is potentially accessible to consciousness whereas procedural knowledge is not. Taken together, the distinction between declarative knowledge and procedural skills is well supported in many ways.

Procedural knowledge can be highly efficient once it has been developed and can work independently without involving declarative knowledge in many cases (e.g., Anderson, 1982; 1983; 1993; Dreyfus and Dreyfus, 1987). However, declarative knowledge is also advantageous in many situations: Declarative knowledge can speed up the learning process when constructed on-line during skill learning (Mathews et al 1989, Sun et al 1996, 1996), declarative knowledge can facilitate the transfer of a skill (Willingham et al 1989, Sun and Peterson 1997, 1998, 1998b) by speeding up learning in new settings, and declarative knowledge can help in the communication of knowledge and skills to others.

On-Line Learning. As demonstrated by Dominowski (1972), Dominowski and Wetherick (1976), Medin et al. (1987), Nosofsky et al (1994) and others, human learning, is often gradual, on-going, and concurrent (with task performance). Although their data and theories are mostly concerned with concept learning, the point applies equally well to skill learning. This view has been implemented in existing top-down learning models such as Anderson (1982, 1983) and Rosenbloom et al (1993). Thus, our attempts at modeling skill learning must also be able to capture this feature. To allow an individual to learn *continuously* from on-going experience in the world, both procedural and declarative knowledge should be acquired in such a gradual, on-going and concurrent way.

Bottom-Up Learning. As mentioned earlier, 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. In Anderson (1982), proceduralization is accomplished by converting explicit declarative knowledge from instruction into production rules, which are subsequently refined through practice. In Anderson (1993), this is accomplished by maintaining explicit memory of instances, which are utilized in performance through analogical processes, and by creating production rules from these instances after repeated use. In Rosenbloom et al (1993), the equivalent of proceduralization is accomplished through "chunking" (i.e., combining production rules; see section 4.3 for details). In Jones and VanLehn (1994), procedural skills are developed through modifying conditions of a priori given rules based on statistical information collected during practice (whereby no procedural/declarative distinction is made; see also Drescher 1989). These models have been applied to a range of domains, for example, in skill learning of theorem proving, text editing, LISP programming, arithmetic, and many other tasks. 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 may 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, Schraagen 1993, Posner et al 1997). 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 control tasks (Berry and Broadbent 1988), artificial grammar learning tasks (Reber 1989), and serial reaction tasks (Willingham et al 1989). Berry and Broadbent (1984) argue that the psychological data in dynamic control 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. Nissen and Bullemer (1987) and 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 (Stanley et al 1989). Using a dynamic control task, Stanley et al. (1989) finds that the development of declarative knowledge paralleled but lagged behind the development of procedural knowledge. [1]

[1] The distinction between implicit and explicit learning is not universally accepted and there are alternative explanations for these findings (Shanks and St.John 1994). However, the distinction does seem to hold up for low-level cognitive

Reber and Lewis (1977) makes a similar observation. Even in high-level *cognitive skill acquisition* (VanLehn 1995), Schraagen (1993) reports that learning in the domain of designing psychological experiments often involves generalizing specific knowledge to form generic schemas/rules (which is, in a sense, bottom-up) in addition to specializing general knowledge to specific knowledge. Rabinowitz and Goldberg (1995) shows that there can be parallel learning at the declarative and procedural levels, separately.

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 provide 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 is not entirely consistent with the top-down approach, because the process can be non-verbal and non-explicit (without awareness; William 1977) and lead to forming action sequences without a priori explicit knowledge. [2] It should be pointed out that conditioning may be applied to simple organisms as well as humans (Thorndike 1927, Wasserman et al 1993, Gluck and Bower 1988). In *developmental psychology*, Karmiloff-Smith (1986) proposes 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. [3]

In sum, there are data and theories available that indicate that learning can proceed from procedural to declarative knowledge (as well as the reverse). Thus, the study of bottom-up skill learning can be justified on both empirical and theoretical grounds.

2.2 The CLARION Model

An Outline of the Model and its Motivations. Reber (1989), based on psychological data, hypothesized that the primary difference between declarative and procedural knowledge lies in the form of their representations. Declarative knowledge is represented explicitly and thus consciously accessible whereas procedural knowledge is represented implicitly and thus inaccessible (Anderson 1983, Ackerman 1988, Reber 1989, and LaDoux 1992). This view is incorporated in our model.

tasks (e.g., Lewicki et al 1992) and thus pertinent to the present work.

[2] We want to point out that although instrumental conditioning without awareness has been controversial, it has nevertheless been demonstrated in laboratory settings under appropriate conditions (e.g., Williams 1977).

[3] This idea can be traced back to Piaget's idea of *restructuring*, and has also been proposed in various forms by other developmental psychologists (e.g. Vygotsky 1986, Keil 1989, and Mandler 1992).

First of all, in our model, the features of procedural knowledge can be captured by a "subsymbolic" distributed representation such as that provided by a backpropagation network (Rumelhart et al 1986), because representational units in a distributed representation are capable of accomplishing tasks but are generally uninterpretable and subsymbolic (see Sun, 1994). [4] (A symbolic representation could be used to represent procedural knowledge, but this would require an artificial assumption that some symbolic representations are not accessible while other similar representations are accessible. Such an assumption seems arbitrary and is not intrinsic to the media of representations. See e.g. Cleeremans 1997 or Sun 1997 for more analyses.) Procedural representation can be modular; that is, a number of small backpropagation networks can exist with each adapted to a specific modality, task, or input stimulus type. This is consistent with the well known modularity claim (Fodor, 1983; Karmiloff-Smith, 1986; Cosmides and Tooby, 1994), and is also similar to Shallice's (1972) idea of a multitude of "action systems" competing with each other.

The learning of procedural knowledge can be captured in a number of different ways. In the learning setting where correct input/output is provided, straight backpropagation (a supervised learning algorithm) can be used for each network. Such supervised learning procedures require the a priori determination of a uniquely correct output for each input. In the learning setting where there is no input/output mapping externally provided, reinforcement learning (in the AI sense, as developed by Sutton 1990, Watkins 1989 and others) can be used. This is preferred in skill learning, because often there is no uniquely correct action although feedback is usually available. Using reinforcement learning, we can measure the goodness of an action through a payoff/reinforcement signal, ranging from, say, 1 to -1 (with 1 being extremely good and -1 being extremely bad and many other possibilities in between). An adjustment can be made to some weights to increase the chance of selecting the actions that receive positive reinforcement and to reduce the chance of selecting the actions that receive negative reinforcement. The process can be based on local weight updating.

In contrast, declarative knowledge can be captured in computational modeling by a symbolic or localist representation (Clark and Karmiloff-Smith 1993), in which each unit has a clear conceptual meaning (i.e., a semantic label). The links connecting such nodes consequently have clear meanings also. This captures the property of declarative knowledge being accessible (through crisp representations in individual nodes) and declarative inferences being performed explicitly (through crisp, articulable links connecting crisp representations of concepts) (Smolensky 1988, Sun 1994). [5] Declarative knowl-

[4]There are alternative views regarding explicitness and verbalizability of knowledge in backpropagation networks. Explanations also vary according to domains. These different views do not directly contradict our view because we allow the possibility of indirect verbalization of knowledge in the networks through some more elaborate transformation process. See section 3.5 for more analyses of this issue.

[5]For arguments that more than distributed representation is needed for modeling explicit processes, see e.g. Cleeremans and McClelland (1991), although they themselves adopted distributed representation (for modeling implicit learning but not explicit learning).

edge can be learned in a variety of ways. This learning is different from the learning of procedural knowledge. Because of the representational difference, one-shot learning is performed. We mainly learn through utilizing procedural knowledge acquired in the bottom level and acquire declarative knowledge on that basis (i.e., the bottom-up learning; Stanley et al 1989, Reber and Lewis 1977, Siegler and Stern 1998). As with procedural knowledge, we dynamically acquire a representation and modify it as needed, to reflect the dynamic on-going nature of skill learning tasks.

This hypothesis of the two different learning processes is consistent with some highly plausible interpretations of relevant findings in the literature (see Sun et al 1996). Berry and Broadbent (1988) demonstrated this difference using two dynamic control tasks that differed only in the degree to which the pattern of correct responding was salient to the subjects. Results suggested that subjects learned the two tasks in different ways: Subjects in the non-salient condition learned the task implicitly while subjects in the salient condition learned the task more explicitly (as measured by tests of resultant explicit declarative knowledge). Reber (1989) described a similar situation in artificial grammar learning. When complex hierarchical relations were needed in order to judge grammaticality, subjects tended to use implicit learning (improving performance but without generating explicit declarative knowledge). When only pair-wise relations were needed, subjects were more likely to use explicit learning through inducing an explicit rule. Mathews et al (1989) essentially demonstrated the same point as well. Although there may be other possible explanations for these findings, it is plausible that the differences above reflect the contrast between two separate learning processes.

The differences in representation and learning between declarative and procedural knowledge lead naturally to "two-level" architectures (see Sun and Bookman 1994, Sun 1995), which consist of two main components: the top level encodes explicit declarative knowledge and the bottom level encodes implicit procedural knowledge (in neural networks as prescribed before). Many existing theories support such a two-level architecture (see Anderson 1993, Bower 1996, Reber 1989, Logan 1988, Posner and Snyder 1975, Seger 1994, Sun 1995).

The relative contributions of the two levels (in learning or performance) may be manipulated experimentally to some degree. Performance can be affected when an agent is influenced to engage in more or less explicit processing. For instance, when an individual is forced to be more explicit, his/her top-level mechanisms may be more engaged and thus the performance may be enhanced to some extent or unaffected depending on circumstances (the effect may be similar to that of a deeper level of processing; Craik and Lockhart 1972; see also Challis et al 1986, Stanley et al 1989, Willingham et al 1989, Gagne and Smith 1962, Ahlum-Heath and DiVesta 1986, Squire and Frambach 1990, Sun et al 1996). When an individual is forced to be overly explicit, his/her top-level mechanisms may be fully engaged but the bottom-level mechanisms may be hampered because of that (as has been observed by, e.g., Reber 1989; see section 3 for details) and thus the performance may be worsened (Reber

1989, Schooler et al 1993, Berry and Broadbent 1984; see section 3). On the other hand, when an individual is distracted by a secondary (explicit) task, his/her top-level mechanisms may be less active in relation to the primary task (i.e., he/she becomes more implicit, because this manipulation affects explicit processes more than implicit processes; Stadler 1995, Nissen and Bullemer 1987, Szymanski and MacLeod 1996; see section 3).

In addition to the two levels, we also include an instance (or episodic) memory which stores recent experiences in the form of "input, output, result" (i.e., stimulus, response, and consequence) that are recency-filtered (see section 3.2 for further details of this memory used in experiments). This memory can also be used for learning. [6] This view of separate instance (episodic) memory and declarative knowledge differs from that of Bower (1996) and Anderson (1993), in both of which instance (episodic) memory is included as part of declarative knowledge.

Here is a summary of the basic model hypotheses:

- Representational difference: The two levels employ two different types of representations and thus have different degrees of accessibility.
- Learning difference: Different learning methods are used for the two levels, although learning is online for both levels.
- Bottom-up learning: When there is no sufficient a priori knowledge available, the learning is bottom-up.
- Separability: The relative contributions of the two levels (in learning or performance) can be experimentally manipulated and thus studied (by using the dual task or the verbalization manipulation).
- A separate instance (episodic) memory.

These hypotheses together form our basic model (Sun et al 1996, Sun 1997). See Figure 2.

The Implementation of the Model. A high-level algorithm that describes the model CLARION, which stands for *Connectionist Learning with Adaptive Rule Induction ON-line*, is as follows:

1. Observe the current state x.
2. Compute in the bottom level the "value" of each of the possible actions (a_i's) associated with the 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 perceptual information x (which goes up from the bottom level) and the existing action rules in place at the top level.

[6]The hypothesis is that the use of instance memory may enhance learning because it provides (at least) additional opportunities for practicing.

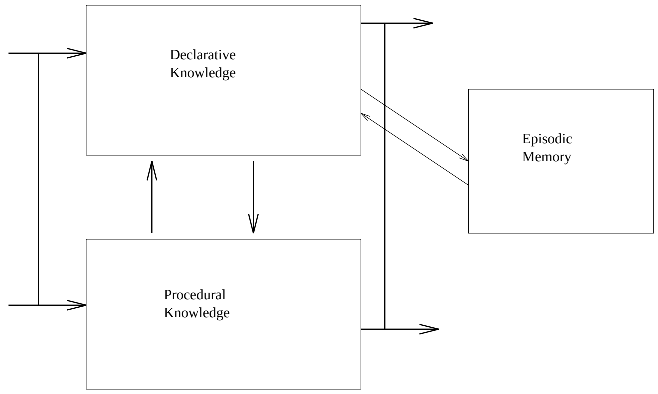

Figure 2: The CLARION architecture.

4. Choose an appropriate action a stochastically, based on combining the values of a_i's (at the bottom level) 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 (to be explained later), based on the feedback information.
7. Update the top level using the *Rule-Construction-Refinement* algorithm (for constructing, refining, and deleting rules, to be explained later).
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. At each step, given the input x, we first compute the Q-values for all the possible actions (i.e., $Q(x, a)$ for all a's). We then use the Q-values to decide probabilistically on an action to be performed, which is done by a Boltzmann distribution of Q-values:

$$p(a|x) = \frac{e^{Q(x,a)/\alpha}}{\sum_i e^{Q(x,a_i)/\alpha}} \tag{1}$$

Here α controls the degree of randomness (temperature) of the decision-making process. (This method is also known as Luce's choice axiom (Watkins 1989). This method is cognitively well justified: it is found to match psychological data in a variety of domains.)

The calculation of Q-values for the current input with respect to all the possible actions is done in a connectionist fashion through parallel spreading activation and is thus highly efficient. Such spreading of activation is assumed to be implicit as, e.g., in Hunt and Lansman (1986), Cleeremans and McClelland (1991), and Bower (1996). We use a four-layered connectionist network (see Figure 3), 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 network is internally subsymbolic and uses implicit representation in accordance with our previous considerations of representational forms. The output of the third layer (i.e., the output layer of the

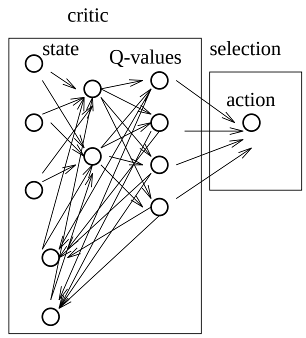

Figure 3: The Q-learning method implemented in a connectionist network. The first three layers constitute a backpropagation network. The output nodes produce Q-values. The fourth layer performs stochastic decision making from all the Q-values.

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 the Boltzmann distribution.

To acquire the Q-values, supervised and/or reinforcement learning methods may be applied (as mentioned earlier). 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:

$$\max(\sum_{i=0}^{\infty} \gamma^i r_i) \tag{2}$$

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 none). The updating of $Q(x, a)$ is based on minimizing

$$r + \gamma e(y) - Q(x, a) \tag{3}$$

where γ is a discount factor, y is the new state resulting from action a in state x, and $e(y) = \max_b Q(y, b)$. 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-values , the agent can learn to take into account future steps in longer and longer sequences. [7]

[7] In terms of both simplicity and performance, Q-learning is the best among similar reinforcement learning methods (Lin 1992, Sun et al 1996). Some learning convergence proofs can be found in Watkins (1989).

Applying Q-learning, the training of the connectionist backpropagation network is based on minimizing the following error at each step:

$$err_i = \begin{cases} r + \gamma e(y) - Q(x,a) & \text{if } a_i = a \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where i is the index for an output node representing the action a_i. Based on the above error measures, the backpropagation algorithm is applied to adjust internal weights (which are randomly initialized before training). (Or, when a correct input/output mapping is available for a step, backpropagation can be directly applied using the error measure of the desired output minus the actual output.) 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 (through temporal difference updating), so that the agent knows which action in a sequence leads to success or failure. This learning process (using Q-learning plus backpropagation) enables the development of procedural skills potentially solely based on the agent independently exploring a particular world on a continuous and on-going basis (without a priori knowledge).

In the top level (see Figure 2), declarative knowledge is captured in a simple propositional rule form. To facilitate correspondence with 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, Towell and Shavlik 1993), in accordance with our previous considerations. Basically, we translate the structure of a set of rules into that of a network. Assume that an input state x is made up of a number of dimensions (e.g., $x_1, x_2, ..., x_n$). Each dimension can have a number of possible values (e.g., $v_1, v_2, ..., v_m$). [8] Rules are in the following form: *current-state* $\longrightarrow$ *action*, where the left-hand side is a conjunction of individual elements each of which refers to a dimension x_i of the (sensory) input state x, specifying a value or a value range (i.e., $x_i \in [v_i, v_i]$ or $x_i \in [v_{i1}, v_{i2}]$), and the right-hand side is an action recommendation a. (Alternatively, the rules can be in the forms of *current-state* $\longrightarrow$ *action new-state* or *current-state action* $\longrightarrow$ *new-state*.) Each element in the left-hand side is represented by an individual node. For each rule, a set of links are established, each of which connects a node representing an element in the left-hand side of a rule to the node representing the conclusion in the right-hand side of the rule. If an element is in a positive form, the link carries a positive weight w; otherwise, it carries a negative weight $-w$. Sigmoidal functions are used for node activation (which is an obvious choice; other functions are also possible):

$$\frac{2}{1 + e^{-(\sum_i i_i w - \tau)}} \tag{5}$$

The threshold τ of a node is set to be $n * w - \theta$, where n is the number of incoming links (the number of the elements in the condition leading to the conclusion represented by this node), and θ is a parameter,

[8]Each dimension is either ordinal (discrete or continuous) or nominal. In the following discussion, we focus on ordinal values; nominal values can be handled similarly. A binary dimension is a special case of a discrete ordinal dimension.

selected along with w to make sure that the node has activation above 0.9 when all the elements of its condition are satisfied, and has activation below 0.1 when one or more elements of its condition are not met. Activations above 0.9 are considered 1, and activations below 0.1 are considered 0. So rules are in fact discretized and thus crisp (binary). [9]

Among other algorithms that we developed, we devised an algorithm (the *Rule-Construction-Refinement* algorithm) for learning declarative knowledge (rules) using information in the bottom level, to capture a bottom-up learning process. The basic idea of this algorithm is as follows: If an action decided by the bottom level is successful (i.e., if it satisfies a certain criterion) then the agent constructs a rule (with its action corresponding to that selected by the bottom level and with its condition specifying 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 constructed rule by considering the outcome of applying the rule: if the outcome is successful, the agent may try to generalize the condition of the rule to make it more universal ("expansion"); if the outcome is not successful, then the condition of the rule should be made more specific and exclusive of the current state ("shrinking"). [10] Specifically, we do the following for *Rule-Construction-Refinement*:

1. Update the rule statistics (to be explained later).
2. Check the current criterion for rule construction, expansion, and shrinking (to be detailed later):
2.1. If the result is successful according to the current criterion, and there is no rule matching that state and that action, then perform *construction* of a new rule: state $\longrightarrow$ action. Add the constructed rule to the rule network.
2.2. If the result is unsuccessful according to the current criterion, revise all the matching rules through *shrinking*:
2.2.1. Remove the matching rules from the rule network.
2.2.2. Add the revised (shrunk) version of the rules into the rule network.
2.3. If the result is successful according to the current criterion, then generalize the matching rules through *expansion*:
2.3.1. Remove the matching rules from the rule network.
2.3.2. Add the expanded rules to the rule network. [11]

[9] In addition, if there is more than one rule that leads to the same conclusion, an intermediate node is created for each such rule: all of the elements on the left-hand side of each rule are linked to the same intermediate node, and then all the intermediate nodes are linked to the node representing the conclusion. For more complex rule forms including predicate rules and variable binding, see Sun (1992). Such rules can be learned using more complex ILP techniques (see Lavrac and Dzeroski 1994).

[10] In this algorithm, specialization and generalization is done based on actually encountered situations, and thus search is minimal. The algorithm is a depth-first specific-to-general search algorithm according to Mitchell (1982). The learning is implemented symbolically.

[11] We also *merge* rules whenever possible: If either rule construction, shrinking or expansion is performed at the current step, check to see if the conditions of any two rules are close enough and thus if the two rules may be combined: If one rule is covered completely by another, put it on the children list of the other. If one rule is covered by another except

Let us discuss the details of the operations used in the above algorithm (including rule construction, shrinking, and expansion) and the criteria measuring whether a result is successful or not (which are used in deciding whether or not to apply some of these operators). The criteria were mostly determined by an *information gain* measure (which compares the quality of two candidate rules).

To calculate the information gain measure, we do the following. At each step, 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 (in Step 1 of the above algorithm) the positive and 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), denoted as 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 1997, 1998b). Each statistic is updated with the following formula: $stat := stat + 1$ (where $stat$ stands for PM or NM); at the end of each episode, it is discounted by: $stat := stat * 0.90$. [12] 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 possible input states), then the rule is considered successful (for the purpose of deciding on expansion or shrinking operations). [13]

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. [14]

for one dimension, produce a new rule that covers both.

[12] The results are time-weighted statistics, which are useful in nonstationary situations.

[13] This is a commonly used method, especially in Inductive Logic Programming, and well justified on the empirical ground; see e.g. Lavrac and Dzeroski (1994).

[14] Potentially, we might use attention to focus on fewer input dimensions, although such attention is not part of the current model.

On the other hand, the criterion for applying the *expansion* and *shrinking* operators 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 a matching 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. [15] Reset all the rule statistics. Any rule covered by the expanded rule will be placed in its children list. [16]

- *Shrinking*: if $IG(C, all) < threshold2$ and $\max_{C'} IG(C', C) > 0$, where C is the current condition of a matching 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. [17] 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.

Note that although the accumulation of statistics is gradual, the acquisition and refinement of rules is one-shot and all-or-nothing.

In the overall algorithm, Step 4 is for making the final decision on which action to take at each step by incorporating outcomes from both levels. Specifically, we combine the corresponding values for an action from the two levels 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

[15] Here $argmax_x f(x)$ is the standard notation that denotes the value of the parameter x that maximizes the function f.

[16] 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.

[17] We should have $threshold2 \leq threshold1$ to avoid oscillation. In our experiment, we set $threshold = 0.08$, $threshold1 = 4.0$, and $threshold2 = 1.0$.

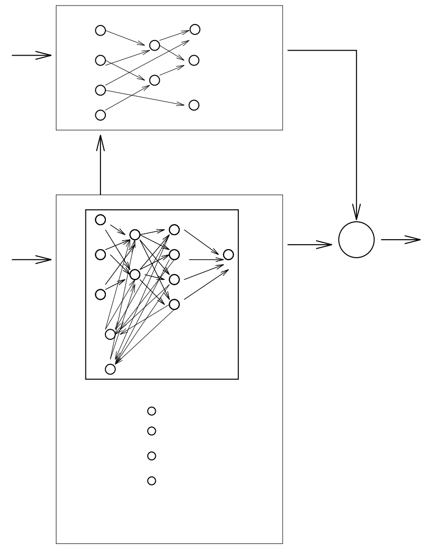

Figure 4: The implementation of CLARION. The top level contains localist encoding of propositional rules. The bottom level contains backpropagation networks. The information flows are indicated with arrows.

decision making with Boltzmann distribution (based on the weighted sums) is then performed to select an action out of all the possible actions. w_1 and w_2 are automatically determined through probability matching (periodically). To justify this cognitively, as shown by Willingham et al (1989), generic declarative knowledge can influence procedural performance in humans. It allows different operational modes: e.g., relying only on the bottom level, relying only on the top level, or combining the outcomes from both levels weighing them differently. These operational modes roughly correspond to the folk psychological notions of the intuitive (reactive) mode, the deliberative mode, and mixtures of the two.

Figure 4 shows the two levels of the model. The combination of multiple learning algorithms in this model is similar to numerous machine learning models/systems that utilize heterogeneous learning algorithms and techniques (such as Maclin and Shavlik 1994, Gelfand et al 1989, Sun and Bookman 1994, Sun and Peterson 1997, and multi-module approaches involving combining decision trees, backpropagation networks and other algorithms). However, in the present work, the justification for such a combination is cognitive, rather than empirical or mathematical as is the case with most hybrid learning models.

3 Comparative Study

We will present below a comparison between human data and model data. We assessed the performance of the model against human performance, with respect to several cross-level manipulations. The comparison provides some support for the model, although the human data do not uniquely support this model. This comparison is the first step toward verifying and developing this generic model. (We did not verify the learning process at the level of individual rules or decisions, because of the serious methodological difficulty discussed in the next section.)

3.1 Experimental Methods

We will describe the experiments with human subjects first. In all of the experiments, subjects were seated in front of a computer monitor that displayed an instrument panel containing several gauges that provided current information (see Figure 5). At the bottom of the screen, the sonar gauges showed how close the mines were at each step, in 7 local areas in front of the vessel that the subjects were supposed to steer, ranging from 45 degrees to the left of the vessel to 45 degrees to the right. Right above the sonar gauges, the bearing gauge showed the direction of the target from the present heading of the vessel. Above and to the left, the fuel gauge showed how much time was left before fuel ran out. To the right was the range gauge, which showed the distance to the target. Subjects used a joystick to navigate the vessel and decided (1) how much to turn and (2) how fast to move. Subjects were given 25 seconds (equivalent to 200 steps), so they were under severe time pressure. The subjects could either (a) reach the target (a success), (b) hit a mine (a failure), or (c) run out of fuel (a failure). 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 8 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).
>
> III. Here is the way to control this submarine. (Experimenter: explain the use of the joystick.)

The above sparse instruction was necessary to inform subjects about the basic setting (including

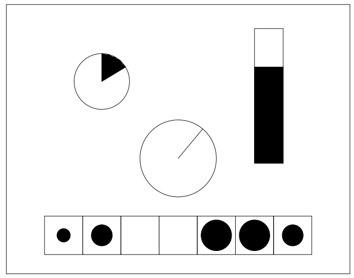

Figure 5: The Navigation Input

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.

the goal of the task). But to encourage bottom-up learning, we avoided providing detailed information about *how* to accomplish the task. Because of severe time pressure and the sparseness of the instruction, subjects had to be reactive in decision making. It was unlikely that subjects had time for instance (episodic) memory retrieval during decision making. However, subjects might use instance (episodic) memory off-line to continue learning (e.g., through memory rehearsal and consolidation).

A random mine layout was generated for each episode. Thus subjects were forced to take into account varying environments and learn only general rules of navigation rather than rules that were specific to a particular layout. This setting is *stochastic*: because subjects can only see a small segment in front, they cannot predict exactly what will turn out (e.g., whether there will be a mine in front or not). The setting is also *non-Markovian*: what will likely turn out may be dependent stochastically on what happened earlier. Five training conditions were used:

- 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 to navigate the minefield.

- The verbalization training condition. This condition was identical to the standard training condition except that subjects were asked to step through slow replays of selected episodes and to verbalize what they were thinking during the episode. [18] 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. They could take as long as they wanted to step through the episodes.

- The over-verbalization training condition. In this condition subjects were required to perform verbalization on 15 of the 25 episodes that they received during one session of training. Replay of an episode occurred immediately after the subject finished the episode. (The exact episodes

[18]Concurrent verbalization was not possible, because only extremely limited time was allowed for each episode.

replayed varied across subjects. However, this manipulation had no discernible effect and was not used in the analysis.)

- The dual-task condition. Subjects performed the navigation task while concurrently performing a category decision task. In the category decision task, subjects were read 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 the dual task).

The rationale for the design of these experiments was to manipulate training settings so as to allow differential emphases on the two levels in human subjects, in order to better understand the interaction of the two levels in them. The effect of these manipulation can be predicted as follows: (1) With verbalization, subjects may be forced to be more explicit, and thus their top-level mechanisms more engaged and their performance enhanced (to some extent, as shown by Stanley et al 1989, Willingham et al 1989). (2) With over-verbalization, subjects were forced to be overly explicit and their top-level mechanisms may be even more engaged, which may hamper the bottom level, and thus their performance may be worsened (as demonstrated in Reber 1989, Schooler et al 1993, Berry and Broadbent 1984). [19] (3) When subjects were distracted by a secondary explicit task in the dual task condition, their top-level mechanisms may be less available to the primary task (because such a secondary task affects explicit processes more than implicit processes; see Stadler 1995, Nissen and Bullemer 1987, Szymanski and MacLeod 1996), which led to worsened performance (Dienes and Fahey 1995, Nissen and Bullemer 1987). (4) When subjects were trained under the dual task condition (even to the same criterion level as under the standard training condition), their transfer performance will be worsened (because learning is more implicit under the dual task condition and implicit learning leads to more limited transfer; see, e.g., Dienes and Berry 1997).

3.2 Model Setup

The CLARION model was specifically tailored to the setting of the above experiments. Our analysis of the task led us to believe that there were three kinds of learning processes: motor learning,

[19] Here the model predicts that there is an inverted U curve. However, we cannot predict precisely the shape of the curve (i.e., how much verbalization is too much) a priori, without matching human data. No model ever can. For example, there is no such a priori prediction in relation to the well-known inverted U curve of learning verb past tense.

perceptual learning, and response selection learning (Proctor and Dutta 1995). Based on subjects' verbal comments and the fact that most subjects had played video games with joysticks extensively, we determined that the motor learning component was negligible. Thus, we omitted motor learning from the current model. Perceptual learning appeared to reflect the learning of which gauge to attend to at various stages in the performance of the task. Perceptual focusing was treated as part of subjects' overall decision making process. In CLARION, this process was carried out at both levels, at the bottom level through (learned) differential weight settings (i.e., input units that acquired high weights through learning matter more than those that acquired low weights — the focus was implicit), and at the top level through (generalized) rule conditions that identified relevant dimensions. Response selection learning involved learning to decide what to do at each moment. This was assumed to be the main part of subjects' decision making process, which was carried out at the two levels of CLARION. In this work, we modeled performance on the basis of the decision making process. [20]

As input, in CLARION each gauge was represented by a set of nodes that corresponded to what human subjects would see on screen. In the discrete input case, generally each node represented one value of a gauge. The number of nodes used to represent each gauge was as follows: One node was used to for "fuel" (which has two values: *a lot* and *a little*), one node for "range" (which has two values: *far* and *near*), six nodes for "bearing" (which represent *far left, left, straight ahead, right, far right,* and *right behind* respectively), and five nodes for each of the seven sonars (each ranging from *very far* to *very close*). This input setup yielded a total of 43 primary perceptual inputs. Thus, there were more than 10^{12} possible input states. [21] [22]

Also in correspondence to the human experimental setting, the action outputs consisted of two clusters of nodes (Maguire et al 1998): one cluster of five nodes represented five different values of "turn" (*left, slightly left, straight, slightly right*, and *right*), and the other cluster of four nodes represented four different values of "speed" (*very fast, fast, slow*, and *standstill*). [23]

Reinforcements for an agent were produced from two sources. One was the gradient reward, which

[20] We did not attempt to model response time. The current interface of the simulation software did not allow us to model response time (Gordon et al 1994).

[21] Thus the model had to deal with the problem of high dimensionality. As a result, a lookup table implementation for Q-learning at the bottom level was not possible (Tesauro 1992, Lin 1992). A functional approximator such as a backpropagation network must be used. The setting is comparable in complexity to many important reinforcement learning applications in AI such as Tesauro (1993) and Mahadevan and Connell (1992).

[22] We tried both discrete and analog input encoding. In the analog case, each gauge was represented by one node that accepted real-valued inputs (which were used in the bottom level; the discretized inputs as described earlier were still used for the top-level rules). We compared the discrete input encoding with the analog one and did not find any significant performance difference.

[23] We used two sets of outputs in the model: one for situations in which the agent was not currently in the minefield and the other for situations in which the agent was in the minefield.

was proportional to the change in the distance to the target. [24] The other was the end reward, which was determined by how successful the agent was at the end of an episode. The end reward was 1 if the agent reached the target within the allotted time, and was inversely proportional to the distance (from the target) if the agent ran out of time or exploded. [25] These reinforcement signals are simple and can be justified cognitively in terms of the fact that human subjects were aware of the necessity of moving closer to the target, avoiding explosion, and reaching the target.

Although some have identified serious problems that may result from a non-Markovian setting and proposed solutions for them (for example, by using a recurrent backpropagation network that uses hidden layers as memory states; Whitehead and Lin 1995), we believe that a feedforward network can work despite the stochasticity and non-Markovianness. The CLARION model used in the experiments was a simplified version in which one feedforward backpropagation network was used (without multiple modules and without recurrent connections). Q-learning was used to perform reinforcement learning at the bottom level. In the top level, rules were in the form of "state $\longrightarrow$ action". Seven hidden units were used in the backpropagation network, the learning rate α was 0.03, the momentum parameter was 0.7, network weights were randomly initialized between -0.01 and 0.01 (thus, Q-values were randomly initialized as well because they were calculated based on network weights), the Q-value discount rate γ was 0.95, and the randomness parameter for stochastic decision making τ was set at 0.01. We set $threshold = 0.08$, $threshold1 = 4.0$, and $threshold2 = 1.0$.

Since the effect of the dual task was mainly in hampering top-level activities (see e.g. Stadler 1995, Nissen and Bullemer 1987, Szymanski and MacLeod 1996). this effect was captured in CLARION by increasing rule learning thresholds so that less activities could occur at the top level. The rule construction threshold *threshold* was increased to 0.15, and the rule expansion threshold $threshold1$ to 6.0. [26] The effect of (regular) verbalization was posited (Stanley et al 1989, Gagne and Smith 1962, Ahlum-Heath and DiVesta 1986, Sun et al 1996) to stem from heightened explicit (rule learning) activities and to a lesser extent, from rehearsing previous episodes/instances. Thus for the model, we used reduced rule learning thresholds (to encourage more rule learning) and also instance memory

[24] When the agent is going toward the target, the reinforcement is $gr = 1/c * ((x_2 - x_1)/x)^4$, where $c = 7.5$, $x_2 - x_1$ is the distance traveled in the target direction in one step, and x is the maximum distance possible (which is 40). When the agent is going away from the target, the reinforcement is $gr' = -0.5gr$.

[25] When the agent runs out of time, the reinforcement is $er = 500/(500 + x) - 1$, where x is the distance to the target. When the agent gets blown up, the reinforcement is $er = 1/2 * 500/(500 + x) - 1$. This reward measures how successful the agent is by how close it managed to get to the target, regardless of what happened at the last step (an explosion, e.g. can be attributed solely to the mistake committed at the last step and has little to do with what happened before that).

[26] Alternatively, we can achieve the same effect through adding noise (which was also a common approach; Cleeremans and McClelland 1991) at the top level. Noise was added in the following way: In calculating the activation of each node at the top level (which was either 0 or 1), there was a 50% chance of a random value being adopted as its activation. The results of these two alternatives were practically the same.

replay (Sutton 1990). The rule construction threshold *threshold* was reduced to 0.06, and the rule expansion threshold *threshold*1 to 2.0. An instance memory of size 200 was used; each encountered step has a 60% probability of being entered into the memory, with random replacement; at each step, a randomly selected step from the instance memory was used for training agents. [27] To capture the effect of over-verbalization, it was hypothesized earlier that too much verbalization caused an agent to engage in overly heightened rule learning activities (that is, it *in effect* switched to a completely explicit mode of learning, as a result of the over-engagement of the top level; Reber 1989, Schooler et al 1993, Berry and Broadbent 1984). This hypothesis was adopted in CLARION: To further encourage more rule learning at the top level, we reduced the rule construction threshold *threshold* to 0, the rule expansion threshold *threshold*1 to 1.0, and the rule shrinking threshold *threshold*2 to -2.0.

The model started out with no more a priori knowledge about the task than a typical human subject, so that bottom-up learning could be captured. The bottom level contained randomly initialized weights (within a fixed backpropagation network topology). The top level started empty and contained no a priori knowledge about the task, either in the form of instructions (Anderson 1982) or instances (Anderson 1993). The interaction of the two levels was not determined a priori either: there was no fixed weight in combining outcomes from the two levels (the weights were automatically set based on relative performance of the two levels on a periodic basis). There was no pre-designed decision-making hierarchy or other similar structures (cf. the hierarchies of problem spaces in Rosenbloom et al 1993, and hierarchies of reinforcement learners in Singh 1994), because we considered it unlikely that such hierarchies were built-in in humans and more likely that they emerged through interacting with the world. The instance memory was empty at the beginning (alternatively, it might contain some information provided in the instruction given to the subjects). There was no supervised learning (i.e., no teacher input). The reinforcement signals were set as described above, which embodied some a priori notions regarding getting close to target and avoiding explosion that were also provided to human subjects through instructions. The learning algorithm with all the requisite parameters was pre-set, presumably reflecting the learning mechanisms of humans. We posited that similar parameters in humans might be tuned by a long evolutionary process that resulted in a near optimal setting. Note that in this set of experiments, these were not free parameters in the sense that they could not be varied to match varying human performance across different training conditions. They were fixed throughout the experimental conditions. [28]

[27] Both mechanisms were employed in an effort to faithfully capture all the major effects of verbalization on human subjects. Performance-wise, instance memory replay was not necessary. We could generate similar performance without such replay.

[28] In other words, they were fixed part of the model specification (not to be adjusted). We varied training conditions of the human subjects and correspondingly the rule learning thresholds of the model to match the subjects, while keeping these parameters fixed. They do not contribute to the degree of freedom that we have in the model to match the *change* of performance across different training conditions by the human subjects.

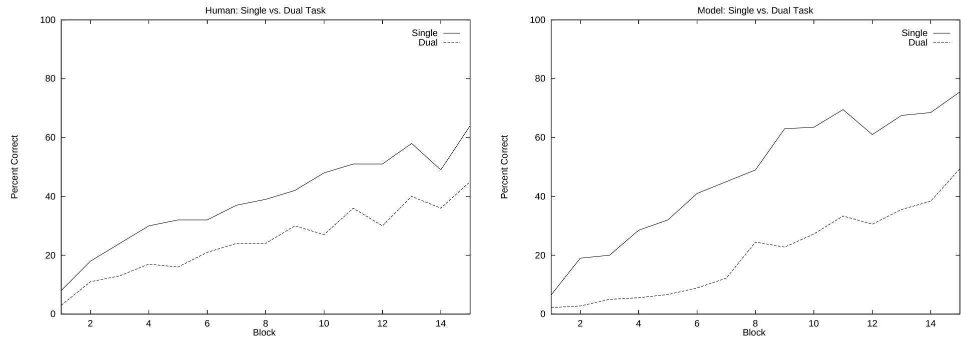

Figure 6: Single vs. Dual Task Training
The left panel contains averaged human data, and the right averaged model data.

3.3 Comparing Human and Model Performance

10 human subjects were compared to 10 model "subjects" in each condition. For model subjects, we randomly set the seeds for the random number generators used (in initializing weights and in decision making), analogous to random selection of human subjects. Thus we did comparisons of two groups of subjects (model vs. human), but not curve fitting in the usual sense. 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. We conjectured that the effect of the dual-task condition was mainly in interfering with the performance of the top-level explicit learning in human subjects (section 2), due to the highly explicit nature of the secondary task, which interfered more with the explicit process than with the implicit process (Stadler 1995, Nissen and Bullemer 1987, Szymanski and MacLeod 1996). Through this manipulation, we expected to obtain a slowed-down learning curve (Nissen and Bullemer 1987, Dienes and Fahey 1995) that reflected more the working of the implicit process when compared with the standard condition. We compared the standard and the dual task condition. In each condition, we obtained performance data over 500 episodes per subject. The 500 episodes were divided into 25 blocks of 20 episodes each. Figure 6 shows the data averaged over the 10 human subjects and the 10 model subjects, respectively (the averaging minimized the impact of individual differences and helped to focus on essential features of learning in this task). To compare human and model performance under single vs. dual task training, 2x2 ANOVA (human vs. model x single vs. dual task) was performed (with success rates averaged for each human or model subject in each condition), which 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. The single task condition led to significantly better performance than the dual task condition. The results support the point that the explicit process and declarative knowledge at the top level help to improve learning.

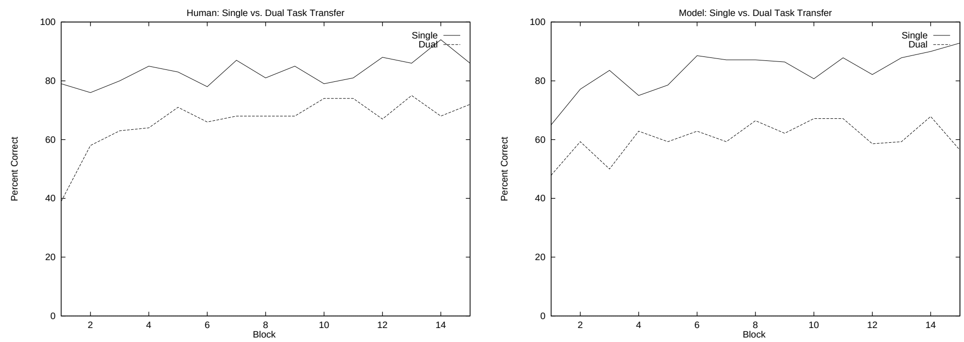

Figure 7: Single vs. Dual Task Transfer

The left panel contains averaged human data, and the right averaged model data.

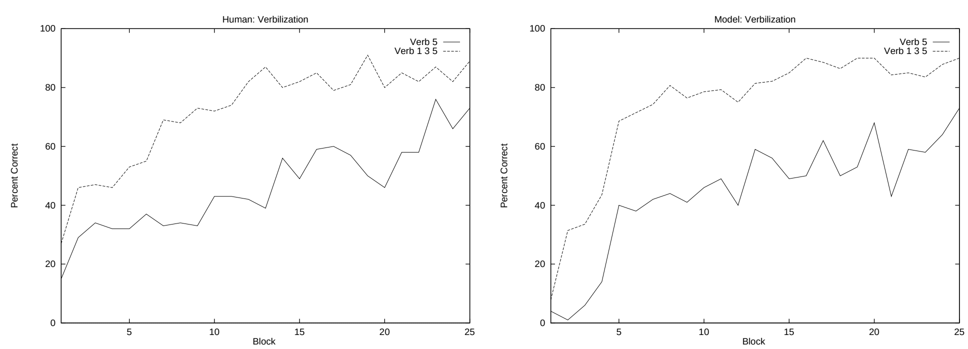

Figure 8: Verbalization vs. No Verbalization

The left panel contains averaged human data, and the right averaged model data.

The effect of the dual task condition on transfer. To further explicate the effect of the dual task condition, we compared the transfer performance after training with the standard vs. the dual-task condition. We hypothesized that declarative knowledge obtained could help to improved the transfer of learned skill (as mentioned in section 2). Thus we expected the dual-task transfer to be worse than single-task transfer. The comparison of the average CLARION and human curves under these two conditions is shown in Figure 7. Our analysis showed that single task transfer was significantly better than dual task 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, indicating similar effects of the dual task condition on the transfer of human and model subjects. This comparison supports the point that declarative knowledge obtained at the top level can help to improve the transfer of learned skill.

The effect of verbalization. Verbalization was expected to improve learning, as observed by Stanley et al (1989) in a dynamic control task under a similar verbalization setting, and also consistent with Willingham et al (1989), which showed that the subjects verbalized better performed better too. This is because the verbalization condition, compared with no verbalization, tends to reflect more of the working of the explicit process in the subjects (a higher degree of contribution from the top-level process; Ahlum-Heath and DiVesta 1986, Squire and Frambach 1990, Stanley et al 1989), and can thus lead to an improved learning curve (Sun et al 1996, Sun and Peterson 1998). This effect can be best revealed by comparing the respective performance of the two groups of verbalization subjects: one started verbalization on the first day and the other on the fifth day. [29] In each group, we obtained performance data over 500 episodes per subject. The 500 episodes were divided into 25 blocks of 20 episodes each. Figure 8 shows the average success rates of the human and the model subjects for each block. 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 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 the effects of verbalization for the two groups (human vs. model) was not significant.

The effect of over-verbalization. A more stark contrast to the standard condition was obtained under the over-verbalization condition. Compared with the standard condition, in the over-verbalization condition, virtually all subjects were performing at floor at the end of their 25 episodes

[29]Because the verbalization setting was different from the standard training setting, due to the camera setup and the difference in the instructions given, the comparison between the standard and the verbalization condition was not performed.

of training. [30] This presumably shows a strong interference by the top level on the implicit learning at the bottom level in the subjects (section 2): This is because if the bottom level is working correctly, the learning performance should generally be much better (see the human performance under the dual-task condition, which presumably reflects mostly the bottom-level implicit learning performance). In contrast, the explicit learning process in the subjects cannot learn well such sequential tasks (as, e.g., demonstrated in an extreme case by Lewicki 1991); [31] thus, the much worsened performance can only be accounted for by the effect of the overly engaged top-level process. This is consistent with a similar hypothesis by Stanley et al (1989), for explaining their findings regarding the difficulty their subjects had in learning a dynamic control task after being given instructions that encouraged them to be explicit. Reber (1989) observed similar phenomena in the artificial grammar tasks. Schooler et al (1993) also reported that requiring verbalization impaired subjects' ability to solve problems that required "insight", by forcing them to be overly explicit. CLARION captured this effect through the aforementioned reduction of the rule learning thresholds: as a result of too much rule learning activities in the top level, the model failed to learn, the same way as human subjects in this condition.

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.

In sum, through these manipulations, we experimentally tested the model, especially in terms of the relative contributions of the two levels. Given that it is impossible to completely separate the respective effects of the two levels in human experiments, the above comparisons of the human and model performance serve to illustrate the effects of the two levels in an indirect way, and thus verify the model indirectly.

[30] 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%. We did not run the experiment longer because of the extremely time-consuming nature of this experiment, which was due to an excessive amount of verbalization.

[31] Note that in CLARION, the top-level learning mechanism, when disconnected from the bottom level, has trouble learning sequential tasks, because of its lack of a temporal credit assignment process (comparable in power to Q-learning) and its all-or-nothing learning process. On the other hand, in the bottom level, the distributed network representation and learning process that incorporates gradedness and temporal information can handle complex sequences.

3.4 Verbalization Indicating Bottom-Up Learning

The verbalization data we collected from the subjects (under the verbalization training condition) were consistent, in an informal sense, with our assumption of bottom-up learning being prominent in this task setting, as exemplified by the following segments.

> S: I thought about it after I started doing it. I said, look at me .... look what I'm doing. I didn't start thinking about it until I started doing it. I figured out that it started helping me and that's when I started doing it myself. (subj.38)

The subject at first performed the task on an "instinctual" basis, without conscious awareness of any particular rules or strategies. Gradually, through "doing it" and then looking at the results, the subject was able to figure out the action rules explicitly. The segment suggested implicit procedural learning at the bottom level and the gradual explication of implicitly learned knowledge.

> S: When I started off ...... I didn't understand at all .... I couldn't grasp the whole sonar concept at all. (subj.38)
>
> S: So, basically what I do – not thinking about driving a submarine or mine. (subj.38)

This is an instance of bottom-level implicit decision making: Having a conceptual (explicit) understanding of sonars, submarine, or mines seemed unnecessary to performing the task.

> S: When you get in a situation like this, where there are gaps, it's purely instinctual. (subj.37)
>
> S: That's pretty much I've done the whole game [being instinctual], with the exception of a couple of patterns I've started to recognize. (subj.37)

This segment is a confirmation of the earlier point: Learning and decision making were mainly done implicitly at the bottom level (at the beginning). There also existed the gradual shift from implicit to explicit learning and decision making.

> S: Straight ... going straight ... I veer left ... straight ... straight ... veer left again ... veer to the opening ... [There were 5 pages of this, on the 1st day] (subj.jc)

This particular subject seemed unable to utter anything but action statements. This seemed to be an indication of the lack of conceptual (explicit) thinking in performing this task, and thus the dominance of bottom-level implicit decision making.

S: I moved back and forth – I didn't understand what to do. I was really confused. ......
I do not know what I did here. (subj.s3)
E: But it looks like you learned a crucial way, which you showed over here.

Although the experimenter noticed that the subject learned in "a crucial way", the subject had no explicit knowledge of it. This indicated the separation of the two levels (as in CLARION) and the significant role of bottom-level learning.

In sum, the verbalization by the subjects suggested that some degree of bottom-level (implicit) learning/decision making and gradual bottom-up learning existed. Performance is often determined by implicit procedural learning, which cannot be easily verbalized, while verbalized explicit knowledge is often nonspecific and has relatively minor impact during learning. This is the kind of learning CLARION was meant to capture.

3.5 Further Analysis

Matching model and human trajectories. We noticed that human and model trajectories were similar in many ways. First, some mine layouts were easier than others, for both human subjects and the model. Both groups exhibited the same pattern — that is, easy layouts were easy for both and hard ones were hard for both. This indicated, to some extent, that performance of CLARION and human subjects was the result of similar processes. Second, both groups preferred a straight path (in the direction of the goal) before entering the minefield and after exiting — indicating again a similar approach. Third, both human subjects and the model seldom reverse course. Once an initial trajectory was launched into the minefield, they continued forward [32] until blown up or time ran out. There was a similar bias in both groups.

Matching rule learning with human data. We thus far could not verify in sufficient detail the rule construction and refinement process of the model with that of human subjects in the navigation domain. This is due to the lack of methodologies accurate enough for the type of low-level skill task that we deal with. Although approaches exist for protocol analysis in high-level cognitive skill domains, they are not applicable to our domain, because in low-level domains (such as navigation), more implicit learning is involved and it is much more difficult to verbalize given the subtle instrumental readings and the minutely different actions (which are not very symbolic; see Chan 1992, Dienes and Berry 1997 for discussions of this issue). During verbalization, a great deal of re-wording took place in most subjects that transformed a detailed, partially implicit, instrument-based decision process to a higher-level coarse-grained explanation in natural language terms, which shed very little light on actual

[32] Even when they were blocked by mines, they would fan back and forth looking for a hole rather than turn around.

declarative knowledge. Verbalization is an interpretive process that involves inference, transformation, and intrusion of implicit knowledge (Nisbett and Wilson 1977). Such verbalization does not reveal, in clear enough terms, the distinction between declarative knowledge and *post hoc* restatement of procedural knowledge (in accordance with our notions of declarative and procedural knowledge; section 2.2). Without such a distinction, little analysis can be performed to verify declarative knowledge. [33] In addition, although declarative knowledge is accessible, some part of it may not be accessed during verbalization, due to a variety of factors. In sum, we need better methodologies in order to verify the model in a more detailed fashion (which will be our future objective). In this work, we are focused on the *effect* of verbalization rather than the content of verbalization.

The possibility of a one-level model. Can a one-level model capture the above human data? Although it is conceivable that a one-level model may be designed so as to capture the data, we failed in our experiments to do so. Specifically, we tested the following possibilities using only the bottom level (with the top level removed). First of all, without the top level, the performance of the bottom level is significantly worse than the human performance in the standard condition, thus generating a poorer match with the human data (despite all possible adjustments of parameters). To capture the dual task condition, we added noise, which did generate worsened performance (compared with the simulation data without noise, which corresponded to the standard condition). To capture the oververbalization condition, we used instance memory replay (rehearsing), which led to worsened performance also (through drastically increasing the amount of replay). However, the replay process failed to produce sufficient improvements in the performance of the bottom level for modeling the verbalization condition (that is, replay in this situation is not sufficient to improve model performance in a statistically significant way, compared with the simulation without replay, which corresponded to the standard condition). See *Analysis of Synergy* for analyses of some other possibilities.

Still, it is possible that some one-level models may work. The human data does not unambiguously point to our model. However, it is seldom, if ever, the case that human data can be used to demonstrate the *unique* validity of a model. We need to rely on converging evidence from various sources, including theoretical accounts (such as those outlined in section 2), to justify a model. By such a standard, this model fares well.

Analysis of synergy. Why are there two separate (although interacting) components? There need to be more reasons than simple redundancy (e.g., for the sake of fault-tolerance). We hypothesize that in general there may be a synergy between explicit and implicit learning/performance at the two respective levels. Such a synergy may show up, under right circumstances, by speeding up learning,

[33] Some may claim that this is not a problem because rule learning itself is an interpretive process. But these two processes are different — one is for internal decision making and the other for verbal, inter-personal communication (Nisbett and Wilson 1977).

improving learned performance, and facilitating transfer of learned skills (as demonstrated in the dual task and the verbalization condition). This is because each level has a different make-up and is thus potentially complementary to the other (see the componential analysis later). [34]

There is psychological evidence in support of this hypothesis. In terms of speeding up learning, Willingham et al (1989) found that those subjects who acquired full explicit knowledge learned faster than those who did not have full explicit knowledge in a serial reaction time task. Stanley et al (1989) reported that subjects' learning improved if they were asked to generate verbal instructions for other subjects along the way during learning. That is, a subject was able to speed up learning through an explication process that generated explicit knowledge. Mathews et al (1989) also showed that learning was improved when both implicit and explicit processes were involved.

In addition, in terms of learned performance, Willingham et al (1989) found that subjects who verbalized while performing tasks were able to attain a higher level of performance than those who did not verbalize, probably because the requirement that they verbalized their knowledge prompted the formation and utilization of explicit knowledge. In high-level skill acquisition, Gick and Holyoak (1980) found that good problem solvers could better state rules that described their actions in problem solving. This phenomenon may be related to the self-explanation effect reported in the cognitive skill acquisition literature (Chi et al 1989): Subjects who explained examples in their physics textbooks more completely did better in solving new problems. In all of these cases, it is likely that the explication process and the use of explicit knowledge helped performance.

In terms of facilitating transfer of skills, Willingham et al (1989) obtained some suggestive evidence that explicit declarative knowledge facilitated transfer. They reported that (1) subjects who acquired explicit knowledge in a training task tended to have faster response times in a transfer task; (2) these subjects were also more likely to acquire explicit knowledge in the transfer task; and (3) these subjects who acquired explicit knowledge responded more slowly when the transfer task was unrelated to the training task, suggesting that the explicit knowledge of the previous task might have interfered with the performance of the new task. In high-level domains, Ahlum-Heath and DiVesta (1986) also found that the subjects who were required to verbalize while solving Tower of Hanoi problems performed better on a transfer task after training than the subjects who were not required to verbalize (see also Gagne and Smith 1962, Squire and Frambach 1990).

Such synergy effects are not universal; they depend on particular experimental settings (for example, when there were too much top-level activities due to over-verbalization, there was a detrimental effect instead). However, we did demonstrate that the interaction of the two levels could be profitably explored (given the right settings). We showed that it was also *possible* that CLARION exhibited

[34] Mathews et al (1989) had a similar hypothesis. Cleeremans and McClelland (1991) and Gibson et al (1997) touched upon the need for explicit processes as well.

analogous synergy effects in learning, performance, and transfer (at least in some settings, e.g., by comparing the standard and the dual task condition, or by comparing the standard and the verbalization condition. (For more experiments with the model that clearly demonstrated the synergy effects in the model, see Sun et al 1996 and Sun and Peterson 1997, 1998, 1998 b.) Similar synergy effects were also observed in CLARION in a different task, the maze running task (Sun et al 1996, Sun and Peterson 1998), which further attested to the synergy hypothesis.

Componential Contributions. We analyzed the source of synergy in the model in a variety of ways. First of all, we tested to see if performance improvement could be attributed to the absence of hidden units in the top level (Sun and Peterson 1998b). We tried the removal of hidden units at the bottom level. Our results indicated that in this way the model as a whole or the bottom level alone had great difficulty in learning the task (see Tesauro 1992 for similar results). In relation to this, we also tested to see if making activation functions of the nodes at the bottom level more discrete by increasing the steepness of the sigmoid function and adding a threshold term (as in the top level) could improve the performance. Our results indicated otherwise. On the other hand, when we added hidden units to the top level and made other nodes there less discrete (i.e., turning the top level into a backpropagation network as in the bottom level), the performance deteriorated, which indicated the importance of discrete localist representations at the top level. In addition, simply removing one level or the other from the model also led to much worsened performance (Sun and Peterson 1998b). [35] Taken together, the results indicated the need for complementary representations in the two levels. See Sun and Peterson (1998b) for the detailed and systematic analysis (which, due to its length, will not be repeated here).

We tested to see if one-shot rule learning at the top level was the source of performance improvement. We compared different frequencies of rule learning in CLARION. A readiness parameter *ready* determines how many times a rule construction/refinement criterion (as explained before) has to be satisfied before a rule can be constructed. We varied *ready* from 0 through 4, for either construction or refinement (or both). The results indicated there was no significant performance difference. However, when we extended *ready* to higher values, the performance of the model deteriorated significantly. This indicated that (nearly) one-shot rule learning supplementing backpropagation learning might also be a partial explanation for the synergy.

The synergy also depended on the setting of the rule learning parameters (thresholds). If the parameters were set too high, there would be too little rule learning to be effective (as in the simulation of

[35] The top-level learning mechanism in CLARION had trouble learning sequential tasks without the bottom level, due to the lack of a temporal credit assignment process (comparable in power to Q-learning) and its all-or-nothing learning process. Although the bottom level could handle sequences, it performed worse without the help of the top level due to the inherent inaccuracy of backpropagation networks (i.e., the blurring effect of their generalization abilities, which might be partially alleviated when a crisp top level was added). See Sun and Peterson (1998b)

the dual task condition). If the parameters were set too low (as in the simulation of over-verbalization), too many erroneous rules would be learned that led to worsened performance, instead of synergy.

We also tested to see if the rule learning criterion was the source of the synergy. We tried two other criteria: the amount of direct reward received by an agent at each step (that is, "if $r >$ threshold"), and the maximum Q-value at a state (that is, "if $Q(x,a) > \max_b Q(x,b) - \epsilon$"). The former 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 latter criterion concerns whether the Q-value of an action is close enough to the maximum Q-value in that state, indicating the optimality of the action. Our experimental results showed that adopting either of these two criteria led to significantly worse performance.

We also investigated other possibilities, such as randomness in the bottom level (controlled by the temperature parameter), the gradient reward used, direct action representation at the top level (as opposed to graded Q-value representation at the bottom level), and so on. It was clear that none of these accounted for the power of the model in generating the synergy. Therefore, the explanation rests on the following factors (identified earlier): (1) the complementary representations of the two levels: discrete vs. continuous; (2) the complementary learning processes: one-shot rule learning vs. gradual Q-value approximation; and (3) the proper bottom-up rule learning criterion used in CLARION.

4 Discussions

4.1 Bottom-up vs. Top-down

Let us examine CLARION in terms of the bottom-up and top-down distinction. The advantage of bottom-up learning is that it does not have to rely on existing, externally given, verbally imparted knowledge (cf. Anderson 1983, 1993, Rosenbloom et al 1993). Developing bottom-up learning models broadens the scope of computational modeling of skill learning, which complements existing models.

There may also be top-down processes involved in many kinds of essentially bottom-up skill learning. For example, it may be necessary for a subject to understand instructions (e.g., in the minefield navigation task), to follow examples (e.g., in learning to swim), or to acquire rules of the game (e.g., in learning to play chess or to drive). In such circumstances, it it necessary to interpret instructions, internalize examples, or memorize rules at the start. General world knowledge and other a priori knowledge would likely be used in these processes. Correct and proper a priori knowledge (examples, mental models, or theories) can get an individual started in the right direction, and then bottom-up learning processes as described here can take over. Improper a priori knowledge about a task, though, may get the individual off in the wrong direction and thus hamper subsequent bottom-up learning (as reported by Stanley et al 1989).

To capture these processes, CLARION can make use of a priori knowledge when it is available. When external instruction in the form of rules is given, the model can wire them into the rule network in the proper place and connects them to existing representations. Wiring up external instruction may require operationalization, that is, turning instructions into a form compatible with internal representations. Gordon and Subramanian (1993) described how operationalization of instruction might be performed, using terminological mapping (mapping into the terminology used by the agent), qualitative-to-quantitative mapping, and factual knowledge of task domains. It may also involve the recoding of given knowledge in an internal form (Maclin and Shavlik 1994). Alternatively, supervised learning of the rule network can be performed (Towell and Shavlik 1993), with, e.g., backpropagation for slow tuning of the rules. Furthermore, CLARION can perform *rule assimilation*, the process by which rules (given externally and wired up in the top level) are assimilated into the bottom level and thus become procedural and more effective (Anderson 1982, Dreyfus and Dreyfus 1987, Gelfand et al 1989). Assimilation is done either using explicitly supervised learning in which the top level serves as the teacher or through gradual practice guided by the top-level knowledge.

4.2 One Level vs. Two Levels

One may argue that if a one-level model can account for data (potentially), then there is no need for the second level. Our view is different: Intuitively and commonsensically, there is the distinction between explicit and implicit thinking (or roughly, conscious and unconscious processes). Furthermore, there are numerous well-put philosophical arguments and psychological data that support this point. The best explanation for this well-established distinction is, by nature, a system that has these two corresponding components, rather than a system that is contrived to implement both types of knowledge in a one-level architecture.

In addition, it is not completely clear that one-level models are always simpler than two-level models. It depends on factors such as parameters, manipulations, and internal structural complexity that are needed to get a one-level model to do the same as a two-level model. For example, the synergy effects (Mathews et al 1989 and Sun and Peterson 1998b; see also section 3.5) cannot be easily generated without (at least) two components. There is no known explanation of the synergy effects without two (or more) components, although such explanations can conceivably be constructed with some complex mechanisms. In this case, a two-level model is an inherently simpler way to capture them than constructing a one-level model to handle both types of knowledge through complicated manipulations. Note that structurally speaking, CLARION is a relatively simple two-level model.

In sum, if we can assume that there are indeed two different types of knowledge that need to be captured (which many theorists have suggested to be the case), then a two-level model seems to be essentially the simplest model.

4.3 Implicit vs. Explicit Plans

In addressing sequential decision making in this work, we are mainly concerned with temporal credit assignment using Q-learning, which is a form of implicit "planning": making decisions based on current information in accordance with a "policy" that implicitly takes into account future steps, and such a "policy" is learned through repeated exposure to various possible sequences in prior experience. However, sometimes *explicit* planning as studied in more traditional models may also be needed in order to fully capture human learning and decision making in skill acquisition (cf. Maguire et al 1998). Explicit planning involves mapping out a sequence of steps explicitly ahead of the time, taking into consideration possible interactions of these steps. This has been discovered in the minefield navigation task, as exemplified by the following segments of verbalization:

> S: I could go forward and I wouldn't get hit by the mine until it gets bigger and I can turn a little bit until there was a smaller or no mine in the middle box. (subj.38)
>
> S: I figured out if they're really close ... if I moved just a little bit, eventually one of them is going to disappear and I can shoot through it. (subj.38)
>
> S: Basically, just going straight ahead waiting for clear spots to appear ... and changing my course to get to the clear spot. (subj.38)
>
> S: Ok. I want to start ... I should have started breaking back to the right, but instead what I think I did, if this is one of my trials, it was so wide open I thought I could get around the back, shoot forward and then come back around. (subj.jc)
>
> S: I'm just thinking – ok I'm trying to find the smallest mine in the middle and I'm trying to go toward it and that's .... so as you can see right here, I found it and I'm starting to go towards it. As it got bigger, I would start turning to where there is no mine at all. (subj.38)

These segments showed several basic elements of explicit planning. They involve plans, goals, and predictions. For example, complex plans were devised: "get around the back, shoot forward and then come back around", or "find the smallest mine in the middle ... go toward it ... as it got bigger ... start turning". Predictions of consequences of actions were also made: a mine "is going to disappear", or a subject is "waiting for clear spots to appear". The goals for actions were also verbalized: "I am trying to find the smallest mine". (Note, however, that these segments are from retrospective verbalizations and thus may involve post hoc rationalization. Thus the existence of explicit planning is yet to be fully explored. As shown by Gardner and Rogoff (1990), explicitness of planning can vary according to settings and instructions.)

Clarion does not yet incorporate such explicit planning. This might partially explain some performance differences observed between the model and the human subjects. However, explicit planning is compatible with the basic framework of Clarion: The bottom level not only provides the necessary information for the construction of rules but it also provides the probabilistic information with regard to what is the best rule to apply in each situation. Constructed rules that include the prediction of new states can be used in forming explicit plans through an explicit search of possible sequences, guided by the probabilistic information.

4.4 Comparisons

Skill Learning. A number of models have been proposed in low-level skill domains such as navigation, dynamic control, and sequence learning. John et al (1994) modeled learning as well as expert performance in a Nintando game with production rules (with the subgoaling and chunking mechanism in SOAR, as will be discussed later; Rosenbloom et al 1993). A large amount of prior knowledge was required in the model. Gelfand et al (1989) proposed a model for robot skill learning that codes all the knowledge in an explicit form and through practice the knowledge is assimilated into a neural network (using backpropagation). In the end, the network is able to capture in an implicit procedural form the skill for performing the task (see also Gordon and Subramanian 1993). Schneider and Oliver (1991) described a hybrid connectionist model that learns skills for judging logic gates. A deliberate calculation is performed first but through repeated trials, an automatized (implicit) process takes over. In these models, skill learning is presented as a top-down process.

There are also many models developed for higher-level skill domains (i.e., cognitive skill acquisition, which is often also concerned with carrying out a sequence of steps in accomplishing a certain goal; VanLehn 1995). Most of the high-level skill learning models are also top-down and rule-based. One of the earliest investigated domains is chess, which has been studied by Herbert Simon and associates based on search in a state space in which means-ends strategies are used. Another frequently explored domain is learning elementary arithmetic, in which Anderson (1982) proposed that learning amounts to turning declarative knowledge obtained from instructions and examples into arithmetic procedures that could be readily used (see also Fitts and Posner 1967). Some of these approaches adopt the distinction between declarative and procedural knowledge, but they mainly focus on learning from instructions/examples and turning them into procedural skills. This focus may reflect the symbolic nature of the chess and arithmetic domains. There are even some indications that such domains are not susceptible to implicit learning.

There are also instance-based theories of skill learning (Logan 1988, Stanley et al 1989), which generally have more of a bottom-up flavor. For example, Logan (1988) showed that skill learning (or more specifically, the automatization of skilled performance) can be captured by the acquisition of domain specific past instances in individuated representational forms (Hintzman 1986). When a situation is encountered, relevant past instances are retrieved (implicitly) and a response is selected based on the instances. However, when a novel situation is encountered where there is no relevant instance, explicit inferences can be performed to select a response. Stanley et al (1989) also described implicit learning/performance as mainly the result of relying on (implicit) memory of past instances. These instances were utilized by comparing a current situation with them and through similarity-based analogical processes they were transformed into a response to the current situation. At first glance, these theories may seem at odds with CLARION. However, upon closer examination, we see that backpropagation networks used in the bottom level of CLARION can be either exemplar-based (i.e., essentially storing instances; see Kruschke 1992) or prototype-based (i.e., summarizing instances). The similarity-based analogical processes alluded to in these theories can be performed (to some degree) in these backpropagation networks, which have been known to excel in similarity-based processes. The instance-based theories, however, do not handle the learning of generalized rules (beyond specific instances).

There are existing models that do acquire more than just instances and learn skills from scratch. But they are not bottom-up in the sense used here because they only learn implicit knowledge. For example, there are a number of connectionist models of implicit learning of sequences. Cleeremans and McClelland (1991) used a recurrent network model to capture qualitatively variance in human data of sequence learning, taking account of gradually widening temporal contexts, priming, and attention effects. Dienes (1992) used a simple one-layer network with either Hebb or Delta learning rules, and performed detailed modeling of sequence learning to obtain a match with human data. These models and results do not easily generalize to bottom-up skill learning in general, though.

There are models that learn more explicit rules but they are usually not cognitively motivated. Grefenstette (1992) and Schultz (1991) developed a model SAMUEL for learning navigation in a simulated two-dimensional world, using GA-based search to find reactive rules. Maes and Brooks (1990) and Sutton (1990) developed reinforcement based models for learning similar skills. Shrager (1990) dealt with "instructionless learning" (which was cognitively motivated), but he used a large amount of a priori knowledge to begin with (see also learning in SOAR as discussed later).

The process difference in the two levels of CLARION resembles to some extent those in some other psychological models of learning: for example, (1) the difference between the strategies of look-ahead vs. table look-up as proposed by Broadbent et al (1986) for explaining the difference between explicit and implicit learning; (2) the difference between algorithms vs. instance retrieval as proposed by Logan

(1988) for accounting for the difference between initial trials and later skillful performance in the course of skill learning; (3) the difference between mental models/theories vs. experiences as proposed by Stanley et al (1989) in relation to the difference between explicit verbalization and implicit skill; (4) the difference between central processing vs. spreading activation proposed by Posner and Snyder (1975) to handle conscious and unconscious memory processes (see also Hunt and Lansman 1986 for a similar view related to attention); (5) the difference between episodic type-2 associations and pre-existing type-1 associations proposed by Bower (1996) for explaining dissociated memory measures. The former type in each dichotomy is more deliberate and more crisp, while the latter is more vague and more automatic. It appears that such a mechanistic difference may be applicable in modeling a variety of cognitive processes. It has been applied in *CONSYDERR* (Sun 1994), a two-level model for everyday commonsense reasoning using a combination of rule-based and similarity-based processes. Another model applying the dual representation framework was *CONSPIRE* (Sun 1994b), which was designed to investigate introspective reasoning through the interaction of two levels via a variety of top-down and bottom-up connections. CLARION is a further extension of these models.

Concept Learning Models. We also compare CLARION with successful psychological models of concept learning such as Nosofsky (1994) and Ahn and Medin (1992). A commonality is that their idea of rule-plus-exception is essentially the same as the rule-plus-network architecture of CLARION. One obvious difference compared with our bottom-up approach to skill learning is that these models are designed for learning that involves a small number of features (or dimensions) and cannot deal with sequential skills in any obvious way.

Our approach also has some remote resemblance to AI work on rule learning and concept formation (for example, ID3 in Quinlan 1986 and AQ in Michalski 1983). However, as "batch" algorithms, they are not directly usable for on-line bottom-up learning; they require teacher input which is not available here. In addition, they do not perform temporal credit assignment and thus cannot handle sequential skill learning tasks. Some incremental, unsupervised learning models, such as COBWEB (Fisher 1987), also differ from our approach in that (1) we have sparse feedback available although it may be delayed, and (2) we usually do not have a complete description of each instance to base decisions on (because of partial observability).

Recently there have been many models combining multiple heterogeneous learning algorithms or techniques for supervised concept learning, such as combining decision trees, backpropagation networks and other algorithms. Although these approaches bear some similarity to the present work, they are aimed at different tasks and they are not cognitively motivated. See Sun and Peterson (1998, 1998b) for more comparisons.

Cognitive Architectures. CLARION can be compared to existing cognitive architectures. In CLARION, as in many of these architectures, different modules are incorporated and elaborate mechanisms are developed. Similar to some architectures, CLARION integrates both symbolic and connectionist methods. Somewhat different from these architectures is the fact that CLARION utilizes a combination of reinforcement-backpropagation learning and rule induction, and exploits synergy of the two. In addition, in CLARION, representations (both explicit and implicit ones) are acquired through autonomous exploration by the learner, instead of being externally given.

Let us compare CLARION with a few architectures in detail. (Note that each of these architecture has advantages that our approach is not designed to capture, although we will only focus on their limitations.) ACT* and ACT-R (Anderson 1982, 1983, 1993) utilize the distinction between procedural and declarative knowledge. ACT* is made up of a semantic network (for declarative knowledge) and a production system (for procedural knowledge). Productions are formed through "proceduralization" of declarative knowledge, modified through use by generalization and discrimination (i.e., specialization), and have strengths associated with them which are used for firing. ACT-R is a descendant of ACT*, in which procedural learning is limited to production formation through mimicking and production firing is based on log odds of success. CLARION succeeds in explaining two issues that ACT does not address. First, while ACT relies mostly on top-down learning (from given declarative knowledge to procedural knowledge), CLARION can proceed completely bottom-up (from procedural to declarative knowledge); CLARION is able to learn on its own without an external teacher providing correct exemplars or instructions of any form. Second, in ACT both declarative and procedural knowledge are represented in an explicit, symbolic form (i.e., semantic networks plus productions), and thus it does not explain, from a representational viewpoint, the differences in accessibility between the two types of knowledge. CLARION accounts for this difference based on the use of two different forms of representations. The top level of CLARION is symbolic/localist and thus naturally accessible/explicit, while the bottom level contains knowledge embedded in a network with distributed representations and is thus inaccessible/implicit. Thus, this distinction in CLARION is *intrinsic* instead of *assumed* as in ACT. [36]

The SOAR (Rosenbloom et al 1993) architecture is based on the ideas of problem spaces, states, and operators. When there is an outstanding goal on the stack, different productions propose different operators and operator preferences for accomplishing the goal. Learning consists of *chunking*, the creation of a new production that summarizes the process leading to achieving a subgoal, so as to avoid impasses subsequently (a form of explanation-based learning). SOAR does not distinguish between the two types of knowledge; chunking is used to account for skill improvement. In terms of the difference in conscious accessibility, it (arbitrarily) assumes the inaccessibility of the working of individual productions, so as to distinguish deliberate and automatized processing with the difference

[36] The newer versions of the ACT models (Anderson 1993) posited a dichotomy of exemplars vs. rules (cf. Logan 1988, Hintzman 1986). However, in these models, exemplars are assumed to be explicit and production rules implicit, which is the opposite of CLARION.

of multiple productions vs. a single production. Also, SOAR requires a large amount of initial (a priori) knowledge about operators and their preferences to begin with; hence the process of learning is not bottom-up.

Drescher (1991) developed an architecture that attempted to implement the Piagetian constructivist view of child development. The learning mechanism is based on statistics collected during interaction with the world. New schemas (i.e., rules) are created and their conditions identified and tuned through statistical means based on relevance (see also Jones and VanLehn 1994). It can also build abstractions out of primitive actions. However, the model does not make the dichotomous distinction of procedural vs. declarative knowledge and thus does not account for the distinction of implicit vs. explicit learning. The model deals only with low-level procedural learning (motor interaction).

5 Conclusions

In this work, we discussed an approach to bottom-up skill learning, and a hybrid connectionist model as a demonstration of the approach. The model consists of two levels for capturing both procedural and declarative knowledge and allows the acquisition of procedural knowledge prior to or simultaneous with the acquisition of declarative knowledge, which differs markedly from most existing models. We believe that the implicit learning and performance at the bottom level is primary and essential, at least in low-level skill domains, and the explicit processes at the top level is secondary. Furthermore, the implicit/explicit difference is explained by the representational difference between the two levels of the model (or in other words, the implicit/explicit difference is reduced to the symbolic/subsymbolic difference in the model and thus they are no longer separate dichotomies).

The model we developed suggests the possibility of alternative ways of skill learning different from existing accounts, and highlights an approach that has been largely neglected. Comparisons with existing learning models (in AI and psychology) showed that our approach has some unique characteristics that other approaches or models did not capture, most notably the bottom-up development and continuous interaction of declarative and procedural knowledge (different from top-down learning, and beyond separate, parallel learning of the two types of knowledge). In addition, we experimentally compared the model and human performance in the minefield navigation task, which demonstrated the relative contributions of the two levels in an indirect way through various experimental manipulations. The match between the model and the human data is significant (although the data does not unambiguously support the model). The shortcoming of the work is the lack of a more detailed match with human subjects, which needs to be addressed in the future with the development of better experimental methodologies.

For more details of the CLARION model, see http://www.cecs.missouri.edu/ ~ rsun/clarion.html

Acknowledgements: This work is supported in part by Office of Naval Research grant N00014-95-1-0440. We wish to thank Susan Chipman and Helen Gigley for the support, Jeff Shrager, Jack Gelfand, Dave Waltz, and Diana Gordon for comments or discussions, and Jim Ballas and Alan Schultz for assistance with the simulator. We would also like to thank the referees for their comments and suggestions. Human experiments were carried out in part by Walter Carr, Toma Ozari, and Sandra Willis.

A Rules learned by CLARION

Below is a set of rules which were obtained at the end of a training session of 500 episodes on the navigation task. Note that we had to re-word (and aggregate) the rules to avoid the use of sonar readings and so on in order to make the rules intelligible. We hypothesized that a similar re-wording process occurred in subjects during verbalization (although the exact nature of it was unknown).

The rules can be aggregated into 3 groups:

- The first group recommends to turn "slight right" if the bearing is straight or slightly right and the right side has a low mine density.
- The second group recommends turning right if the target is straight or slightly right and the right side has the lowest mine density.
- The third group recommends going left if the target is straight to left and the lowest density side is the left.

In this particular run, incidently, no rule concerning going straight is learned. "slight right" acts as "straight" — in actuality, they are very similar. This "slight right" preference may lead to drifting right over the long run. The drift is corrected/compensated by the left turn policy in the third group of rules (because this group of rules is more general and thus very likely applied). Overall, the three groups of rules, especially the first and the third, work together to avoid hitting mines while going in the general direction of the target.

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.

P. Agre and D. Chapman, (1990). What are plans for? In: P. Maes, ed. *Designing Autonomous Agents.* Elsevier, New York.

M. Ahlum-Heath and F. DiVesta, (1986). The effect of conscious controlled verbalization of a cognitive strategy on transfer in problem solving. *Memory and Cognition.* 14, 281-285.

W. Ahn and D. Medin, (1992). A two-stage model of category construction. *Cognitive Science.* 16, 81-121.

J. R. Anderson, (1976). *Language, memory, and thought.* Erlbaum, Hillsdale, NJ.

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

J. R. Anderson, (1983). *The Architecture of Cognition*, Harvard University Press, Cambridge, MA

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

A. Barto, S. Bradtke, and S. Singh, (1996). Learning to act using real-time dynamic programming. *Artificial Intelligence.* 72(1), 81-138

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

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

G. Bower, (1996). Reactivating a reactivation theory of implicit memory. *Consciousness and Cognition*, 5, 1/2, 27-72.

D. Broadbent, P. Fitsgerald, and M. Broadbent, (1986). Implicit and explicit knowledge in the control of complex systems. *British Journal of Psychology.* 77, 33-50.

J. Bruner, J. Goodnow, and J. Austin, (1956). *A Study of Thinking.* Wiley, New York.

B. Challis, B. Velichkovski, F. Craik, (1996). Level-of-processing effects on a variety of memory tasks. *Consciousness and Cognition*, 5 (1/2), 142-164.

C. Chan, (1992). *Implicit cognitive processes: Theoretical issues and applications in computer systems design.* Doctoral dissertation, University of Oxford, Oxford, UK.

M. Chi, M. Bassok, and M. Lewis, P. Reimann, and P. Glaser, (1989). Self-explanation: how students study and use examples in learning to solve problems. *Cognitive Science.* 13. 145-182.

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, (1997). Principles for implicit learning. In D. Berry (Ed.), *How implicit is implicit learning?* , pp. 195-234. Oxford (England): Oxford University Press.

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

L. Cosmides and J. Tooby, (1994). Beyond intuition and instinct blindness: toward an evolutionarily rigorous cognitive science. *Cognition.* 50, 41-77.

F. Craik and R. Lockhart, (1972). Level of processing: a framework for memory research. *Journal of Verbal Learning and Verbal Behavior.* 11, 671-684.

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

Z. Dienes, (1992). Conenctionist and memory-array models of artificial grammar learning. *Cognitive Science.* 16. 41-79.

Z. Dienes and D. Berry, (1997). Implicit learning: below the subjective threshold. *Psychonomic Bulletin and Review.* 4, 3-23.

Z. Dienes and R. Fahey, (1995). Role of specific instances in controlling a dynamic system. *Journal of Experimental Psychology: Learning, Memory and Cognition*, Vol.21, No.4, 848-862.

R. Dominowski, (1972). How do people discover concepts? In: R. L. Solso (Ed.), *Theories in Cognitive Psychology: The Loyola Symposium.* Potomac, Md: L. Erlbaum, pp. 257-288.

R. Dominowski and N. Wetherick, (1976). Inference processes in conceptual rule learning. *Journal of Experimental Psychology: Human Learning and Memory.* 2 (1), 1-10.

G. Drescher, (1991). *Made-up Minds.* MIT Press. Cambridge, MA.

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

D. Fisher, (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning.* 2, 139-172.

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

J. Fodor, (1983). *The Modularity of Mind.* MIT Press, Cambridge, MA.

R. Gagne and E. Smith, (1962). A study of the effects of verbalization on problem solving. *Journal of Experimental Psychology*, 63, 12-18.

W. Gardner and B. Rogoff, (1990). Children's deliberateness of planning according to task circumstances. *Developmental Psychology*, 26 (3), 480-487.

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.

F. Gibson, M. Fichman, and D. Plaut, (1997). Learning in dynamic decision tasks: computational model and empirical evidence. *Organizational Behavior and Human Decision Processes*, 71 (1), 1-35.

M. Gick and K. Holyoak, (1980). Analogical problem solving. *Cognitive Psychology.* 12, 306-355.

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

D. Gordon and D. Subramanian, (1993). A multistrategy learning scheme for agent knowledge acquisition. *Informatica.* 17, 331-346.

D. Gordon, A. Schultz, J. Grefenstette, J. Ballas, and M. Perez. (1994). *User's Guide to the Navigation and Collision Avoidance Task.* Naval Research Lab. Washington, DC.

J. Grefenstette, The evolution of strategies for multiagent environments. *Adaptive Behavior.* 1(1). 65-90.

D. Hintzman, (1986). Schema abstraction in a multiple-trace memory model. *Psychological Review.* 93, 528-551.

J. Holland, N. Nisbett, P. Thagard and J. Holyoak, (1986). *Induction: A Theory of Learning and Development.* MIT Press, Cambridge, MA.

E. Hunt and M. Lansman, (1986). Unified model of attention and problem solving. *Psychological Review.* 93 (4), 446-461.

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

R. Jones and K. VanLehn (1994). Acquisition of children's addition strategies: a model of impasse-free knowledge-level learning. *Machine Learning.* 16, 11-36.

A. Karmiloff-Smith, (1986). From meta-processes to conscious access: evidence from children's metalinguistic and repair data. *Cognition.* 23. 95-147.

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

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

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

J. LeDoux, (1992). Brain mechanisms of emotion and emotional learning. In: *Current Opinion in Neurobiology.* Vol.2, No.2, 191-197.

P. Lewicki, T. Hill, and M. Czyzewska, (1992). Nonconscious acquisition of information. *American Psychologist.* 47, 796-801.

L. Lin, (1992). Self-improving reactive agents based on reinforcement learning, planning, and teaching. *Machine Learning.* Vol.8, pp.293-321.

G. Logan, (1988). Toward an instance theory of automatization. *Psychological Review.* 95 (4), 492-527.

R. Maclin and J. Shavlik, (1994). Incorporating advice into agents that learn from reinforcements. *Proc.of AAAI-94.* pp. 694-699, Seattle, WA. Morgan Kaufmann, San Meteo, CA.

P. Maes and R. Brooks, (1990). Learning to coordinate behaviors, *Proc.of National Conference on Artificial Intelligence.* pp.796-802. Morgan Kaufmann, San Mateo, CA.

E. Maguire, N. Burgess, J. Donnett, R. Frackowiak, C. Frith, and J. O'Keefe, (1998). Knowing where and getting there: a human navigation network. *Science*, 280, 921-924.

J. Mandler, (1992). How to build a baby. *Psychology Review.* 99, 4. 587-604.

R. Mathews, R. Buss, W. Stanley, F. Blanchard-Fields, J. Cho, and B. Druhan, (1989). Role of implicit and explicit processes in learning from examples: a synergistic effect. *Journal of Experimental Psychology: Learning, Memory and Cognition.* 15, 1083-1100.

D. Medin, W. Wattenmaker, and R. Michalski, (1987). Constraints and preferences in inductive learning: an experimental study of human and machine performance. *Cognitive Science.* 11, 299-339.

R. Michalski, (1983). A theory and methodology of inductive learning. *Artificial Intelligence.* Vol.20, pp.111-161.

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

R. Nisbett and T. Wilson, (1977). Telling more than we can know: verbal reports on mental processes. *Psychological Review.* 84 (3), 1977.

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

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

M. Posner and C. Snyder, (1975). Facilitation and inhibition. In: *Attention and Performance.* P. Rabbitt and S. Dornick. (eds.) Academic Press.

M. Posner, G. DiGirolamo, and D. Fernandez-Duque, (1997). Brain mechanisms of cognitive skills. *Consciousness and Cognition.* 6, 267-290.

J. Proctor and A. Dutta, (1995). *Skill Acquisition and Human Performance.* Sage Publications, Thousand Oaks, CA.

R. Quinlan, (1986). Inductive learning of decision trees. *Machine Learning.* 1, 81-106.

M. Rabinowitz and N. Goldberg, (1995). Evaluating the structure-process hypothesis. In: F. Weinert and W. Schneider, (eds.) *Memory Performance and Competencies.* Lawrence Erlbaum, Hillsdale, NJ.

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

A. Reber and S. Lewis, (1977). Implicit learning: an analysis of the form and structure of a body of tacit knowledge. *Cognition*, 5, 333-361.

P. Rosenbloom, J. Laird, and A. Newell, (1993). *The SOAR papers: Research on Integrated Intelligence.* MIT Press, Cambridge, MA.

D. Rumelhart, J. McClelland and the PDP Research Group, (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, MIT Press, Cambridge, MA.

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

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

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.

A. Schultz, (1991). Using a genetic algorithm to learn strategies for collision avoidance and local navigation. *Proc. of 7th International Symp. on Unmanned Untethered Submersible Technology.* 213-225. U. of New Hampshire, Durham.

C. Seger, (1994). Implicit learning. *Psychological Bulletin.* 115 (2), 163-196.

T. Shallice, (1972). Dual functions of consciousness. *Psychological Review.* 79 (5), 383-393.

D. Shanks, and M. St. John, (1994). Characteristics of dissociable learning systems. *Behavioral and Brain Sciences*, 17, 367-394.

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

R. Siegler and E. Stern, (1998). Conscious and unconscious strategy discovery: a microgenetic analysis. *Journal of Experimental Psychology: General*, 127 (4), 377-397.

J. Shrager, (1990). Commonsense perception and the psychology of theory formation. In: J. Shrager and P. Langley, (eds.) *Computational Models of Scientific Discovery and Theory Formation.* Morgan Kaufmann Publishers. San Mateo, CA.

S. Singh, (1994). *Learning to Solve Markovian Decision Processes.* Ph.D Thesis, University of Massachusetts, Amherst, MA.

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

L. Squire and M. Frambach, (1990). Cognitive skill learning in amnesia. *Psychobiology.* 18, 109-117.

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

W. Stanley, R. Mathews, R.Buss, and S. Kotler-Cope, (1989). Insight without awareness: on the interaction of verbalization, instruction and practice in a simulated process control task. *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, (1994b). The CONSPIRE architecture. Technical Report, Department of Computer Science, University of Alabama. Tuscaloosa, AL.

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

R. Sun, (1997). Learning, action, and consciousness: a hybrid approach towards modeling consciousness. *Neural Networks*, special issue on consciousness. 10 (7), pp.1317-1331.

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

R. Sun, T. Peterson, and E. Merrill, (1996). Bottom-up skill learning in reactive sequential decision tasks. *Proc.of 18th Cognitive Science Society Conference*, pp.684-690. Lawrence Erlbaum Associates, Hillsdale, NJ.

R. Sun and T. Peterson, (1997). A hybrid model for learning sequential navigation. *Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'97).* Monterey, CA. pp.234-239. IEEE Press, Piscataway, NJ.

R. Sun and T. Peterson, (1998). Some experiments with a hybrid model for learning sequential decision making. *Information Sciences.* Vol.111, pp.83-107.

R. Sun and T. Peterson, (1998 b). Autonomous learning of sequential tasks: experiments and analyses. *IEEE Transactions on Neural Networks*, Vol.9, No.6, pp.1217-1234.

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. pp. 216–224.

K. Szymanski and C. MacLeod, (1996). Manipulation of attention at study affects an explicit but not an implicit test of memory. *Consciousness and Cognition*, 5 (1/2), 165-175.

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

E. Thorndike, (1911). *Animal Intelligence.* Hafner, Darien, Connecticutt.

G. Towell and J. Shavlik, (1993). Extracting Refined Rules from Knowledge-Based Neural Networks, *Machine Learning.* 13 (1), 71-101.

K. VanLehn, (1991). Rule acquisition events in the discovery of problem-solving strategies. *Cognitive Science.* 15, 1-47.

K. VanLehn, (1995). Cognitive skill acquisition. In: J. Spence, J. Darly, and D. Foss, (eds.) *Annual*

Review of Psychology, Vol.47. Annual Reviews Inc, Palo Alto, CA.

L. Vygotsky, (1962). *Thought and Language.* MIT Press, Cambridge, MA.

E. Wasserman, S. Elek, and D. Chartlosh, and A. Baker, (1993). Rating causal relations. *Journal of Experimental Psychology: Learning, Memory, and Cognition.* 19, 174-188.

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

S. Whitehead and L. Lin, (1995). Reinforcement learning of non-Markov decision processes. *Artificial Intelligence.* 73 (1-2). 271-306.

B. Williams, (1977). Verbal operant conditioning without subjects' awareness of reinforcement contingencies. *Canadian Journal of Psychology*, 31 (2), 90-101.

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.