The Interaction of Implicit Learning, Explicit Hypothesis Testing Learning, and Implicit-to-Explicit Knowledge Extraction

Ron Sun
Department of Cognitive Science
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
rsun@rpi.edu

Xi Zhang
Paul Slusarz
Department of Computer science
University of Missouri-Columbia
Columbia, MO 65211

Robert Mathews
Psychology Department
Louisiana State University
Baton Rouge, LA 70803-5501

June 12, 2006

Corresponding author: Ron Sun

Abstract

To further explore the interaction between the implicit and explicit learning processes in skill acquisition (which have been tackled before, e.g., in Sun et al 2001, 2005), this paper explores details of the interaction of different learning modes: implicit learning, explicit hypothesis testing learning, and implicit-to-explicit knowledge extraction. Contrary to the common tendency in the literature to study each type of learning in isolation, this paper highlights the interaction among them and various effects of the interaction on learning, including the synergy effect. This work advocates an integrated model of skill learning that takes into account both implicit and explicit learning processes; moreover, it also uniquely embodies a bottom-up (implicit-to-explicit) learning approach in addition to other types of learning. The paper shows that this model accounts for various effects in the human behavioral data from the psychological experiments with the process control task, in addition to accounting for other data in other psychological experiments (which has been reported elsewhere). The paper shows that to account for these effects, implicit learning, bottom-up implicit-to-explicit extraction, and explicit hypothesis testing learning are all needed.

Keywords: cognitive modeling, cognitive science, psychology, skill learning, implicit learning, neural networks, backpropagation, reinforcement learning

1 Introduction

The role of implicit learning in skill acquisition and the distinction between implicit and explicit learning have been widely recognized in recent years (see, e.g., Reber 1989, Stanley et al 1989, Willingham et al 1989, Proctor and Dutta 1995). Although implicit learning as well as explicit learning have been actively investigated, the complex interaction between the implicit and the explicit and the importance of this interaction have not been widely recognized; such interaction has traditionally been downplayed or ignored, with only a few notable exceptions (e.g., Mathews et al 1989, Sun et al 1998, 2001, 2005). Research has been focused on showing the *lack* of explicit learning in various learning settings (see especially Lewicki et al 1987) and on the controversies stemming from such claims. Similar oversight is also evident in computational simulation models of implicit learning (with few exceptions such as Cleeremans 1994, Sun et al 2001).

Despite the lack of studies of interaction, it has been gaining recognition that it is difficult to find a situation in which only one type of learning is engaged (Reber 1989, Seger 1994, but see Lewicki et al 1987). Various indications of the interaction are scattered throughout the literature. For instance, Stanley et al (1989) found that under some circumstances concurrent verbalization (which leads to generating more explicit knowledge) could help to improve human subjects' (most implicit) performance in a process control task (the detail of which will be explained later). Ahlum-Heath and DiVesta (1986) also found that verbalization led to better performance in learning Tower of Hanoi. (However, note that, as Reber (1976, 1989) and Sun et al (2001) pointed out, verbalization and the resulting explicit knowledge might also hamper implicit learning, especially when too much verbalization induced an overly explicit learning mode in human subjects performing a task that was not suitable for learning in an explicit way.)

As variously demonstrated by Berry and Broadbent (1988), Stanley et al (1989), and Reber et al (1980), verbal instruction given prior to learning can also facilitate or hamper task performance. One type of instruction was to encourage explicit search by human subjects for regularities that might aid in task performance. For example, Reber et al (1980) found that, depending on the ways stimuli were presented, explicit search might help or hamper performance. Owen and Sweller (1985) and Schooler et al (1993) found that explicit search hindered learning. Another type of instruction was explicit how-to instruction that told human subjects specifically how the tasks should be performed, including providing detailed information concerning regularities in stimuli. Stanley et al (1989) found that such instructions helped to improve performance significantly.

In terms of the relation between implicit and explicit knowledge acquired during learning, there is some evidence that implicit and explicit knowledge may develop independently under some circumstances. Willingham et al (1989), for example, reported some psychological data that were consistent with the parallel development of implicit and explicit knowledge. By using two different measures (with varying criterion levels) for assessing the two types of knowledge respectively, they compared the time course of implicit and explicit learning. It was shown that implicit knowledge might be acquired in the absence of explicit knowledge and vice versa.

There are also cases where a subject's performance improves earlier than explicit knowledge can be verbalized by the subject (Stanley et al 1989). For instance, in process control tasks (to be detailed later), while the performance of the subjects quickly rose to a high level, their verbal knowledge improved far more slowly: The subjects could not provide usable verbal knowledge (for novice subjects to use) until near the end of their training (see Stanley et al 1989). It appears that in these tasks, it is much easier to acquire implicit skills than to acquire explicit knowledge, and hence there is a delay in the development of explicit knowledge. Bowers et al (1990) also showed delayed learning of explicit knowledge. When subjects were given certain partial patterns to complete, they first showed implicit recognition of proper completion (though they did not have explicit recognition). Their implicit recognition improved over time until eventually an explicit recognition was achieved. This phenomenon was also demonstrated by Reber and Lewis (1977) in artificial grammar learning. In all of these cases, as suggested by Stanley et al (1989), Seger (1994), and Sun et al (2001), due to the fact that explicit knowledge lags behind but improves along with implicit knowledge, explicit knowledge is in a way "extracted" from implicit knowledge. Learning of explicit knowledge appears to occur through the (delayed) *explication* of implicit knowledge.

In the development of cognitive architectures, the distinction between procedural and declarative knowledge has been proposed for a long time, and adopted by many in the field (e.g., Anderson 1993), although not all (e.g., Rosenbloom et al 1993). The distinction maps roughly onto the distinction between the explicit and implicit knowledge, because procedural knowledge is generally inaccessible while declarative knowledge is generally accessible and thus explicit (but also see differing views in, e.g., Anderson and Lebiere 1998, Lebiere, Wallach, and Taatgen 1998). However, in work on cognitive architectures, the focus has been almost exclusively on *top-down* models (that is, learning first explicit knowledge and then implicit knowledge on the basis of the explicit knowledge), the *bottom-up* direction (that is, learning first implicit knowledge and then explicit knowledge, or learning both in parallel) has been largely ignored. As pointed out earlier, there is work that did demonstrate the parallel

development of the two types of knowledge or the extraction of explicit knowledge from implicit knowledge (e.g, Rabinowitz and Goldberg 1995, Willingham et al 1989, Stanley et al 1989), contrary to the common top-down approaches in developing cognitive architectures.

With regard to the interaction between implicit and explicit processes, many issues arise: (1) How can we best model implicit and explicit learning processes computationally? (2) How is bottom-up learning (implicit-to-explicit transition) possible and how can it be realized computationally (e.g., Stanley et al 1989; Sun et al 2001)? (3) How do these different types of learning (explicit, implicit, and implicit-to-explicit) interact and contribute to overall skill learning? (4) How do different types of knowledge (acquired from these different types of learning) interact during skilled performance and what is the impact of that interaction on performance? For example, the synergy of the two may be produced, as shown by Sun et al (2001).

In order to further understand the interaction between the implicit and explicit learning processes in skill acquisition (Sun et al 2001, Sun and Zhang 2002, 2003), this paper explores finer details of the interaction of different learning modes: that is, implicit learning, learning through explicit hypothesis testing, and implicit-to-explicit knowledge extraction (bottom-up learning). Contrary to the prevailing tendency in the literature to study each type of learning in isolation, this paper highlights the interaction among them and various effects of the interaction on learning, including the synergy effect. This work advocates an integrated model of skill learning that takes into account both implicit and explicit learning processes. Moreover, it embodies both a bottom-up learning approach (first learning implicit knowledge and then explicit knowledge on the basis of implicit knowledge) and an explicit hypothesis testing learning approach towards learning explicit knowledge. The paper shows that this model accounts for various effects of the implicit/explicit interaction demonstrated in the psychological literature.

In this work, we choose to use the human psychological data of the process control task from Stanley et al (1989). The data are typical of human performance in process control tasks, as well as in implicit learning tasks in general, and they demonstrate the interaction between the explicit and implicit learning processes in especially interesting ways. The essential task setting is as follows: Human subjects were instructed to control the output of a simulated system by choosing their input into the system (from a fixed set of available inputs; see examples later on). The output of the system was determined from the input provided by the subjects, through a functional relationship. However, this relationship was not known to the subjects. Subjects learned gradually how to control the output of the system, through trial-and-error, and many of them also developed some explicit knowledge of

the relationship later on. Various experimental manipulations of learning settings placed differential emphases on different types of knowledge in the human subjects. The data as reported by Stanley et al (1989) demonstrated a number of interesting effects of the implicit/explicit interaction, as touched upon earlier (see also Sun et al 2005): (1) the verbalization effect (i.e., verbalization sometimes leads to better performance), (2) the explicit how-to instruction effect (i.e., receiving instructions leads to better performance), and (3) the synergy effect (the enhanced role of explicit processes in the verbalization and the explicit instruction condition leads to better performance). We will show that our model is capable of accounting for these different effects of the implicit/explicit interaction. (In addition, the model has been shown before to be capable of accounting for other psychological data in other tasks as well, the details of which have been reported elsewhere, such as Sun et al 2001, Sun and Zhang 2002, 2003, Sun et al 2005.) We will show later that, to account for these effects, implicit learning, bottom-up implicit-to-explicit extraction, and explicit hypothesis testing learning are all needed.

In the remainder of this paper, we will first introduce the CLARION model and highlight its features relevant for simulating the afore-mentioned data set (in section 2). In section 3, we will present a few simulations of the data set and some analysis. Section 4 presents a general discussion and section 5 some conclusions.

2 The CLARION Model

Let us look into a model that incorporates both implicit and explicit processes, namely CLARION (Sun et al 2001, Sun 2002). CLARION has a dual representational structure — implicit and explicit representations being in two separate "levels" (Seger 1994). Essentially it is a dual-process theory of mind (Chaiken and Trope 1999). It also consists of a number of functional subsystems (including the action-centered subsystem, the non-action-centered subsystem, and so on; Sun 2002).

CLARION is a comprehensive theory about the human mind. As such, it is highly complex, and contains numerous algorithmic or mathematical details. (Note that the details of the model have been described extensively in a series of previous papers, including Sun (1997), Sun and Peterson (1998), Sun et al (1998, 2001), and Sun (2002).) In this work, we will focus only on the action-centered subsystem (the ACS), which is most relevant to the process control task at hand.

The action-centered subsystem (the ACS) consists of two levels of representations: the Implicit

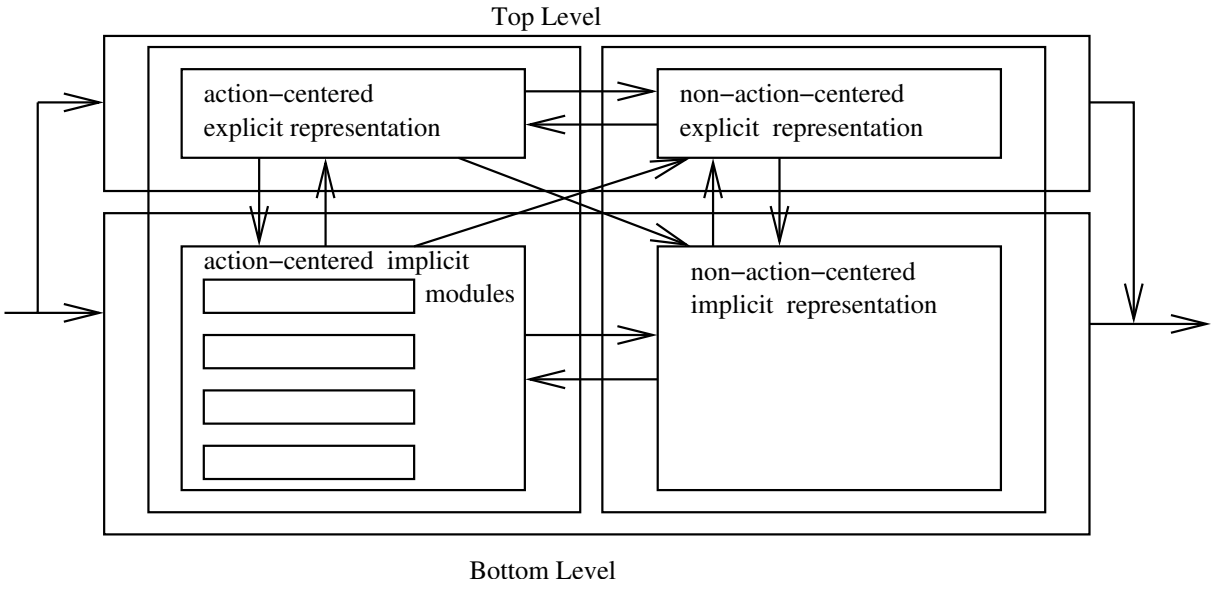

Figure 1: The overview of CLARION. Within the action-centered subsystem (the left side), the top level (the ARN) contains explicit encodings of propositional rules and the bottom level (the IDN) contains backpropagation networks for capturing implicit skills. The information flows are indicated with arrows.

Decision Network (the IDN) in the bottom level and the Action Rule Network (the ARN) in the top level. See Figure 1 (which includes both the action-centered and the non-action-centered subsystem). Let us look into some details of the action-centered subsystem below.

Overall Action Decision Making. First, we will take a look at an overall picture — the overall decision making in the action-centered subsystem. Action decision making is generally as follows (details of which will be explicated subsequently):

1. Observe the current state x.
2. Compute in the bottom level (i.e., the IDN, or the Implicit Decision Network) the "value" of each of the possible actions (a_i's) in the current state x: $Q(x, a_1)$, $Q(x, a_2)$, ..., $Q(x, a_n)$. Stochastically choose one action based on the values.
3. Find out all the possible actions (b_1, b_2, ..., b_m) at the top level (i.e., the ARN, or the Action Rule Network), based on the current state information x (which goes up from the bottom level) and the existing rules in place at the top level. Stochastically choose one action.
4. Choose an appropriate action a, by stochastically selecting the outcome of the top level or the bottom level.

5. Perform the selected action a, and observe the next state.
6. Update the bottom level in accordance with Q-learning (implemented within a backpropagation network; to be detailed later).
7. Update the top level using an appropriate learning algorithm (for constructing, refining, and deleting explicit rules; to be detailed later).
8. Go back to Step 1.

Representation. Let us first consider representations, emphasizing the distinction between implicit and explicit processes. The inaccessible nature of implicit knowledge may be captured by "subsymbolic" distributed representations such as provided by a backpropagation network (Rumelhart et al 1986). This is because representational units in a distributed representation are capable of accomplishing tasks but are generally not individually meaningful (see Rumelhart et al 1986, Sun 1995). This characteristic of distributed representation accords well with the inaccessibility of implicit knowledge. [1] In contrast, explicit knowledge may be captured in computational modeling by a symbolic or localist representations (Clark and Karmiloff-Smith 1993), in which each unit is easily interpretable and has a clearer conceptual meaning. This characteristic captures the property of explicit knowledge being more accessible and more manipulable (Sun 1995). This radical difference in the representations of the two types of knowledge leads to a two-level model whereby each level using one kind of representation captures one corresponding type of process (either implicit or explicit). Sun (1995, 1997, 2002) contains more theoretical arguments for such two-level models, which we will not get into here.

Learning of Implicit Knowledge. Let us turn to the learning of action-centered knowledge (which is most relevant to this work). First, the learning of implicit action-centered knowledge at the bottom level of the ACS (that is, in the IDNs) can be done in a variety of ways consistent with the nature of distributed representations. Reinforcement learning can be used (Watkins 1989), especially Q-learning (Watkins 1989) implemented using backpropagation networks. Such learning methods are cognitively justified (Shanks 1993, Cleeremans 1997, Sun 2002).

Specifically, $Q(x, a)$ is the "quality value" of action a in state x. It is output from a backpropagation network. Actions can be selected based on Q values, for example, using the Boltzmann distribution

[1] However, it is generally not the case that distributed representations are not accessible at all but they are definitely less accessible, not as direct and immediate as symbolic/localist representations. Distributed representations may be accessed through indirect, transformational processes.

(Watkins 1989):

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

where i ranges over all possible actions, and τ controls the degree of randomness in decision making.

In terms of learning Q values, there are a number of possibilities. In the regular Q-learning algorithm, the updating of $Q(x, a)$ is based on minimizing:

$$\Delta Q(x, a) = \alpha(r + \gamma e(y) - Q(x, a)) \tag{2}$$

where x is the current state, a is one of the actions, α is the learning rate, r is the immediate reward, γ is a discount factor, y is the new state resulting from action a in state x, $e(y) = \max_b Q(y, b)$, and b denotes any possible action, 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) total reinforcement to be received if action a is performed, and $r + \gamma e(y)$ estimates the (discounted) total reinforcement to be received, 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) total reinforcement. Using Q-learning allows implicit, reactive sequential behavior to emerge (Sun et al 2001, Sun 2002).

In the simplified Q-learning algorithm, we learn the Q value function as follows:

$$\Delta Q(x, a) = \alpha(r + \gamma \max_b Q(y, b) - Q(x, a)) = \alpha(r - Q(x, a)) \tag{3}$$

where x is the current state, a is one of the actions, α is the learning rate, r is the immediate reward, and $\gamma \max_b Q(y, b)$ is set to zero, because in this case we rely on immediate reward (it amounts to "supervised" learning).

The afore-defined $\Delta Q(x, a)$ provides the error signal needed by the backpropagation algorithm and then backpropagation takes place. That is, backpropagation learning is based on minimizing the following error at each step:

$$err_i = \begin{cases} \Delta Q(x, a_i) & \text{if } a_i = a \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where i is the index for an output node representing the action a_i. Based on the above error measure, the backpropagation algorithm is applied to adjust internal weights (which are randomly initialized before training).

Learning of Explicit Knowledge: the RER Algorithm. The action-centered explicit knowledge at the top level of the ACS (that is, in the ARN) can also be learned in a variety of ways in

accordance with the symbolic/localist representations used. Because of the representational characteristics, one-shot learning based on hypothesis testing (Nosofsky et al 1994, Sun 1997) is needed. With such learning, individuals explore the world, and dynamically acquire representations and modify them as needed, reflecting the dynamic (on-going) nature of skill learning (Sun 1997, Sun et al 2001). The implicit knowledge already acquired in the bottom level can be utilized in learning explicit knowledge, through bottom-up learning (Sun et al 2001). Therefore, we investigate two possible algorithms: one for bottom-up learning (from implicit to explicit knowledge) and the other for independent hypothesis testing learning at the top level.

The RER algorithm (Rule Extraction and Refinement; Sun et al 2001) captures bottom-up learning. 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 extracts a rule (with its action corresponding to that selected by the bottom level and with its condition specifying the current state), and adds the rule to the top level. 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; if the outcome is not successful, then the condition of the rule should be made more specific and exclusive of the current state.

1. Update the rule statistics (i.e., the positive or negative match counts, to be explained below).
2. Check the current criterion for rule extraction, generalization, and specialization (based on the afore-mentioned statistics):
2.1. If the result is successful according to the current criterion, and there is no rule matching the current state and action, then perform *extraction* of a new rule: state $\longrightarrow$ action. Add the extracted rule to the top level.
2.2. If the result is unsuccessful according to the current criterion, revise all the rules matching the current state and action through *specialization*:
2.2.1. Remove these rules from the top level.
2.2.2. Add the revised (specialized) versions of the rules into the top level.
2.3. If the result is successful according to the current criterion, then generalize the rules matching the current state and action through *generalization*:
2.3.1. Remove these rules from the top level.
2.3.2. Add the generalized rules to the top level.

Let us discuss the details of the operations used in the above algorithm and the criterion measuring whether a result is successful or not.

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 (i.e., $PM_a(C)$ and $NM_a(C)$), 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), each denoted as C, with regard to the action a performed. That is, $PM_a(C)$ (i.e., Positive Match) equals the number of times that an input state matches condition C, action a is performed, and the result is positive; $NM_a(C)$ (i.e., Negative Match) equals the number of times that an input state matches condition C, action a is performed, and the result is negative. Positivity or negativity (for updating PM and NM) may be determined based on (delayed or immediate) feedback. For example, the inequality, $r > threshold$, may determine the positivity/negativity of a step and of the rule matching this step.

Based on these statistics, we may calculate the information gain (IG) measure:

$$IG(A, B) = log_2 \frac{PM_a(A) + c_1}{PM_a(A) + NM_a(A) + c_2} - log_2 \frac{PM_a(B) + c_1}{PM_a(B) + NM_a(B) + c_2} \tag{5}$$

where A and B are two different rule conditions that lead to the same action a, and c_1 and c_2 are two constants (the default values are $c_1 = 1, c_2 = 2$). The measure compares essentially the percentages of positive matches under alternative conditions A and B (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 following algorithm, if a rule is better compared with the corresponding match-all rule (i.e, the rule with the same action but with the condition that matches all possible input states), then the rule is considered successful.

We decide on whether or not to extract a rule based on the positivity criterion, which measures whether the current step is successful or not, fully determined by the current step (x, y, r, a):

- *Extraction*: if the current step is positive (according to the current positivity criterion) and if there is no rule that covers this step at the top level, set up a rule $C \longrightarrow a$, where C specifies the values of all the dimensions exactly as in the current input state x and a is the action performed at the current step. [2]

[2]Note that the above (default) extraction method is suitable for specific-to-general rule learning. That is, we extract a most specific rule and then try to generalize it later. An alternative, for general-to-specific rule learning, is to extract a rule with a small condition involving only one input dimension (or a few), and then try to specialize it later.

On the other hand, *generalization* and *specialization* operators are based on the afore-mentioned information gain (IG) measure. Generalization 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 specialization 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. Note that the general idea of this process is similar to many machine learning algorithms from artificial intelligence (e.g., Mitchell 1998), except that here it is done on-line while interacting with the environment. Due to the on-line nature, some details of this process are different from existing algorithms. Here are the detailed descriptions of these two operators:

- *Generalization*: if $IG(C, all) > threshold1$ and $\max_{C'} IG(C', C) \geq 0$, where C is the current condition of a rule, *all* refers to the corresponding match-all rule (with the same action as specified by the original 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 a generalized condition is potentially better**], then set $C'' = argmax_{C'} IG(C', C)$ as the new (generalized) condition of the rule. [3] Reset all the rule statistics. [4]

- *Specialization*: if $IG(C, all) < threshold2$ and $\max_{C'} IG(C', C) > 0$, where C is the current condition of a rule, *all* refers to the corresponding match-all rule (with the same action as specified by the original 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 specialized condition is better**], then set $C'' = argmax_{C'} IG(C', C)$ as the new (specialized) condition of the rule. Reset all the rule statistics. [5] If specializing the condition makes it impossible for a rule to match any input state, delete the rule.

An Example of RER. Let us examine the hypothetical simulation of a process control task. In the task, a simulated system output is presented to subjects and the subjects are instructed to

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

[4] Any rule covered by the generalized rule will be placed in its children list. 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 specialized, some or all of those rules on its children list may be reactivated if they are no longer covered.

[5] Restore those rules in the children list of the original rule that are not covered by the specialized rule and the other existing rules.

provide an input that will lead to the next output of the simulated system being at a target level (Berry and Broadbent 1987). Through continuously interacting with the simulated system, subjects learn, implicitly or explicitly, the relationship between their input to the system and the output from that system.

In the model, learning experience at the bottom level promotes the formation of implicit knowledge. Learning is done through iterative weight updating (the idea of Q-learning implemented with backpropagation was described before). The resulting weights specify a function relating the input to the subjects (which consists of some previous history of interaction with the system) and the output from the subjects (which is the input into the simulated system to be controlled). The input/output function is specified in an implicit way (i.e., embedded in two layers of weights within a backpropagation network). For example, the iteratively updated weights may end up specifying the following relation (embodied in the weights): If the current system output is 4000 and the previous system output was 5000, provide an input of 500 to the system.

The acquired functional knowledge at the bottom level can lead to the extraction of explicit knowledge at the top level. The initial extraction step creates a rule that corresponds to the current input and the output (as determined by the bottom level). The generalization step adds more possible values (that may match new situations) to the condition of the rule so that it may have more chances of matching new situations. The specialization step, on the other hand, adds constraints to the rule (by removing possible matching values in the condition of the rule) to make it more specific and less likely to match other situations. The application of these steps is determined based on the information gain measure described before. For example, the following rule may be initially extracted: If the current system output is 4000, the previous system output was 5000, and the system output before that was 3000, provide an input of 500 to the system. Generalization steps may lead to a simplified rule: If the current system output is 4000 and the previous system output was 5000, provide an input of 500 to the system (provided that the IG measure calculated allows generalization). Further generalization may lead to: If the current system output is 4000, provide an input of 500 to the system. Continued revision (through a mixed sequence of generalizations and specializations) is likely to happen, as determined by the IG measure (which is in turn determined by the performance of the revised rule).

Learning of Explicit Knowledge: the IRL Algorithm. In the IRL algorithm (Independent Rule Learning; Slusarz and Sun 2001), rules are hypothesized and tested in the top level. Initially, we hypothesize rules of a certain form to be tested (e.g., Dienes and Fahey 1995, Nosofsky et al 1994). When a measure of a rule (the IG measure) falls below the deletion threshold, we delete the

1	$P = aW + b$
2	$P = aW + cP_1 + b$
3	$P = aW_1 + b$
4	$P = aW_1 + cP_2 + b$

Figure 2: The order of IRL rules to be tested. $a = 1, 2$, $b = -1, -2, 0, 1, 2$, $c = -1, -2, 1, 2$, P is the desired system output level (the target), W is the current input to the system (to be determined), W_1 is the previous input to the system, P_1 is the previous system output level (under W_1), and P_2 is the system output level at the time step before P_1.

rule. Whenever all the rules of a certain form are deleted, a new set of rules of a different form are hypothesized, and the cycle repeats itself. In hypothesizing rules, we progress from the simplest rule form to the most complex, for example, in the order as shown in Figure 2, in accordance with those numerical relations used in human experiments (Berry and Broadbent 1988, Stanley et al 1989). Other rule forms can be easily added to the hypothesis testing process. Since rules of the same form are tested in a parallel fashion, adding more rules will not drastically change the working of the model.

As before, the IG measure of a rule is calculated based on the immediate reward at every step when the rule is applied. The inequality, $r > threshold$, may determine the positivity/negativity of a step and of the rule matching this step. Then, PM (positive match) and NM (negative match) counts of the matching rule are updated. IG is then calculated based on PM and NM:

$$IG(C) = log_2 \frac{PM(C) + c_1}{PM(C) + NM(C) + c_2} \tag{6}$$

where C is the current rule and c_1 and c_2 are two constants (where the default values are $c_1 = 1, c_2 = 2$). Thus, IG in this case essentially measures the positive match ratio of a rule. We use "$IG(C) < threshold4$" as a criterion for deciding when to delete a rule. (Note that $threshold3$ is used in CLARION to denote something else and is not relevant here.)

Selection between the Outcomes of the Two Levels. In step 4 of the overall algorithm for action decision making described earlier, the probabilities for the selection of either the outcome of the bottom level (the IDNs) or the top level (the ARN) are determined based on the relative performance of the two levels, through "probability matching". That is, the two levels compete against each other to be used in directing actions, based on their respective performance. Specifically, the *success rate* of a component is calculated based on the positive match ratio — how many positive matches a component produces within the context of all the matches. The success rates are then used for

determining selection probabilities of different components (Sun 2002).

The relative contributions of the two levels may be manipulated experimentally (to some degree). Performance can be affected when an individual 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 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 1996, Stanley et al 1989, Willingham et al 1989, Gagne and Smith 1962, Ahlum-Heath and DiVesta 1986, Squire and Frambach 1990). When an individual is forced to be completely 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) and thus the performance may be worsened (Reber 1989, Schooler et al 1993, Berry and Broadbent 1984, Sun et al 2001).

Further Details. Further details of CLARION may be found at the Web site:

http://www.cogsci.rpi.edu/∼rsun/clarion.html.

Various previous simulations of psychological data can also be found there.

3 Simulation of Process Control Data

In this paper, we focus on simulating the process control task of Stanley et al (1989). We are especially interested in capturing the interaction of the different types of learning in the data. The experiments involved various experimental manipulations of learning settings that placed differential emphases on the two levels. These data can be captured using the two-level interactive perspective as embodied in CLARION.

In accounting for the data, we aim to capture (1) the verbalization effect, and (2) the explicit how-to instruction effect, as mentioned before (Sun et al 2005). (Through capturing the verbalization effect and the explicit instruction effect, we at the same time capture the synergy effect, which can be discerned through contrasting these two conditions with the standard condition — both indicate that the enhancement of the top level leads to better performance.) Through the simulations, it will be shown that the division of labor between, and the interaction of, the two levels is important.

For modeling each of these manipulations, only one or a few parameter values were changed. These parameters were changed as follows. To capture the verbalization effect, we changed the thresholds

at the top level. The hypothesis is that, as explained earlier, verbalization tends to increase top-level activities, especially rule learning activities. To capture the explicit instruction effect, we simply encode the given explicit knowledge at the top level.

In the remainder of this section, we present two different simulations. These simulations capture the data of Stanley et al (1989). To demonstrate robustness of the architecture some variations in representations and in details of learning algorithms are involved.

3.1 Stanley et al. (1989): Simulation 1

Task Settings. In Stanley et al (1989), two versions of a process control task were used. In the "person" version, subjects were to interact with a computer simulated "person" whose behavior ranged from "very rude" to "loving" (over a total of 12 levels) and the task was to maintain the behavior of the simulated "person" at "very friendly" by controlling his/her own behavior (which could also range over the 12 levels, from "very rude" to "loving"). In the sugar production factory version, subjects were to interact with a simulated factory to maintain a particular production level (out of a total of 12 possible production levels), through adjusting the size of the workforce (which also had 12 levels). In either case, the behavior of the simulated system was determined by $P = 2 * W - P_1 + N$, where P was the current system output, P_1 was the previous system output, W was the subjects' input to the system, and N was noise. Noise (N) was added to the output of the system, so that there was a chance of being up or down one level (a 33% chance respectively).

There were four groups of subjects. The control group was not given any explicit how-to instruction and not asked to verbalize. The "original" group was required to verbalize: Subjects were asked to verbalize after each block of 10 trials. Other groups of subjects were given explicit instructions in various forms: To the "memory training" group, a series of 12 correct input/output pairs was presented. To the "simple rules" group, a simple heuristic rule ("always select the response level half way between the current production level and the target level") was given. The numbers of subjects varied across groups. 12 to 31 subjects were tested in each group. All the subjects were trained for 200 trials (20 blocks of 10 trials).

Results. The exact target value plus/minus one level was considered on target. The mean scores (numbers of on-target responses) per trial block for all groups were calculated. Analysis showed that the score for the original group was significantly higher than the control group ($F(1, 73) = 5.20, p <$

Human Data

	Sugar Task	Person Task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Figure 3: The human data from Stanley et al (1989). Each data point indicates the number of on-target responses per trial block.

0.05). Analysis also showed that the scores for the memory training group and for the simple rule group were also significantly higher than the control group. See Figure 3. [6]

Model Setup. Only the ACS of CLARION was used in this simulation. In the bottom level, one IDN was used for implementing the regular Q-learning with backpropagation. The reinforcement was determined by whether or not the outcome from the to-be-controlled system was on target (the exact target value plus/minus one). If the outcome was the target value plus/minus one, then the reward was one; otherwise the reward was $0.2 \times (1 - |actual - target|)$. (We also tried a few other reward schemes, and obtained essentially the same results. This particular reward setting was not crucial.)

The input dimensions of the simulated subject included: the target value for the system to be controlled, the serial position t[7], the action at step $t-1$, the output from the system to be controlled at step $t-1$, the action at step $t-2$, and the output from the system to be controlled at step $t-2$. The output of the simulated subject consisted of one dimension of 12 possible actions. The encoding at the bottom level was such that each value in each input dimension was represented by an individual node in the input layer of the backpropagation network. The output encoding at the bottom level used a set of nodes, each for one possible action. Thus, we used 60 input units, 40 hidden units, and 12 output units.

At the top level, both RER and IRL rule learning were involved. Four sets of IRL rules were used. See Figure 2. (Note that other rule forms can be easily added to the hypothesis testing process. Since rules of the same form are tested in a parallel fashion, adding more rules will not drastically change

[6]Note that subjects performed somewhat better in the person task compared with the sugar task. We hypothesize that subjects may have brought in their prior knowledge of interacting with other people in the real world into their performance of this task.

[7]Note that this dimension was excluded from RER.

the working of the model.) Positivity was measured by whether or not the system outcome (sugar production) was on target (the exact target value plus/minus one). (This criterion was adopted based on human experiments — Berry and Broadbent (1988) and Stanley et al (1989) used this criterion to count the number of successful trials.) This criterion was used for calculating PM and NM (for both RER and IRL rules), and also for extracting initial rules in RER. Based on that, the IG measures for RER and IRL, respectively, were used.

Other parameters were as follows: $\alpha = 0.05$, $\gamma = 0.95$, $\tau = 0.09$, $threshold = 1.0$, $threshold1 = 2.0$, $threshold2 = 1.2$. and $threshold4 = 0.2$.

For modeling each of the experimental conditions, only one or a few parameter values were changed. These parameters were changed as follows. To model the effect of verbalization (in the "original" group), we raised the IRL rule deletion threshold at the top level, and lowered the RER rule learning thresholds at the top level. The new thresholds were: $threshold = -0.1$, $threshold1 = 1.0$, $threshold2 = 0.5$. and $threshold4 = 0.5$. [8] The hypothesis was that verbalization tended to increase top-level activities, especially rule learning activities. To capture explicit instructions, we simply wired up explicit a priori knowledge at the top level. In the "memory training" condition, each of the 12 examples was wired up at the top level (in the form of $P_1 \longrightarrow W$). In the "simple rule" condition, the simple rule (as described earlier) was wired up at the top level.

100 runs were conducted for each group. Each run lasted 20 blocks (for a total of 200 trials). To capture the fact that subjects performed somewhat better in the person task compared with the sugar task (due to the fact that subjects brought in their prior knowledge of interacting with other people in the real world into their performance of this task), we conducted some pre-training (for 20 blocks of 10 trials) prior to training on the person task.

Simulation Results. First of all, our simulation captured the verbalization effect in the human data well. See Figure 4. We used t tests to compare the simulated "original" group with the simulated control group, which showed a significant performance improvement ($p < .01$), analogous to the human data.

Our simulation also captured the explicit instruction effect (shown in Figure 4). We used t tests to compare the "memory training" and "simple rule" groups with the control group in the model data, which showed significant improvements of these two groups over the control group ($p < .01$), analogous

[8] Different from RER, with IRL, a higher threshold ($threshold4$) leads to more rule learning activities at the top level.

Human Data

	Sugar Task	Person Task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Model Data

	Sugar Task	Person Task
control	1.92	2.62
original	2.77	4.01
memory training	4.45	5.45
simple rule	4.80	5.65

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+RER+IRL	0.113	0.178	0.048

Figure 4: The simulation of Stanley et al. (1989). Each data point indicates the number of on-target responses per trial block.

to the human data.

To better understand the contributing factors in the model performance, we performed a componential analysis of the model components, to discern the contributions of various constituting components of the model. We did "lesion" studies of the full model and tests of partial models, to discover the respective effects of the top level vs. the bottom level and RER vs. IRL,

We first performed a "lesion" study of the full model. After we optimized the parameters of the full model, we removed IRL or RER respectively to form two partial models. With the same setting of parameters (previously optimized with regard to the full model), we applied the two partial models to the learning of this task. We noticed that removing IRL led to far worse performance than removing RER. That is, there were indications that IRL contributed much more significantly to the performance of the original full model than RER. See Figure 5 for the "lesion" simulation data.

We wondered if this effect was an artifact of our parameter setting, which was optimized with regard to the full model. So we also tested individually optimized partial models, through optimizing parameters (those parameters that were applicable to a partial model) with regard to each partial

Human Data

	Sugar Task	Person Task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Model Data (IDN+RER)

	Sugar Task	Person Task
control	1.55	1.89
original	1.60	1.95
memory training	3.77	4.15
simple rule	4.08	4.45

Model Data (IDN+IRL)

	Sugar Task	Person Task
control	2.10	2.65
original	3.45	4.68
memory training	4.71	5.80
simple rule	5.06	6.29

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+RER	1.231	0.466	1.996
IDN+IRL	0.384	0.485	0.283

Figure 5: The simulation of Stanley et al. (1989) with "lesioned" models. Each data point indicates the number of on-target responses per trial block.

model individually in terms of maximizing the match between the partial model and the human data. (Note that our optimization was done by hand, through trial and error, so we could not guarantee its absolute optimality.) With the same training of 20 blocks of 10 trials each, we compared the match of the resulting partial models with that of the full model (see Figure 6). We noticed that (1) the model with IDN and IRL performed much better than the model with IDN and RER, but (2) neither performed as well as the full CLARION model. This result showed that IRL was more important than RER in matching the human data, but all of three components, IDN, IRL, and RER, were useful in matching the human data, and all were necessary in order to optimize the match between human and model performance. See Figure 6 for all the relevant data.

Furthermore, based on each of these partial models, we built complete models. For each partial model, we first froze all the parameters applicable to the partial model (which were optimized with respect to the partial model), and then added missing components to make the partial model complete, in the meantime optimizing only those parameters that were applicable to those newly added components. The reason for doing so was to further identify the significance of each component through freezing other components.

We compared the resulting "partial-full" models with each other and with that of the original full model. We noticed that adding IRL to the partial model with IDN and RER led to more improvements than adding RER to the partial model with IDN and IRL, which again showed the importance of IRL in this task. See Figure 7 for the data. As before, this test showed that all the above components were useful in terms of producing a better fit with the human data.

Discussions. In all, the simulation and the human data both confirmed the verbalization effect and the explicit instruction effect. In so doing, the match between the two and thus the validity of the CLARION model was also demonstrated.

The model was able to capture these effects in the human data because it used explicit rule learning to supplement implicit learning at the bottom level. Regardless of whether RER or IRL was involved, explicit rule learning captured explicit knowledge in skilled performance, which helped to enhance performance. This simulation demonstrated the usefulness of explicit learning (both IRL and RER) in capturing various effects in human skill learning (Sun et al 2001, 2005). It was also apparent from the data that IRL contributed considerably more to the capturing of the human data than RER. So, the simulation suggested, correspondingly, that it was possible that in human learning of this task, explicit hypothesis testing learning (as captured by IRL) was more important than implicit-to-explicit

Human Data

	Sugar Task	Person Task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Model Data (IDN+RER)

	Sugar Task	Person Task
control	1.68	1.81
original	1.64	1.96
memory training	4.23	4.46
simple rule	4.72	4.87

Model Data (IDN+IRL)

	Sugar Task	Person Task
control	2.23	2.76
original	3.43	4.55
memory training	4.55	5.63
simple rule	4.86	5.63

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+RER	1.016	0.407	1.624
IDN+IRL	0.285	0.387	0.184

Figure 6: The simulation of Stanley et al. (1989) with partial models. Each data point indicates the number of on-target responses per trial block.

Human Data

	Sugar Task	Person Task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Model Data (IDN+RER $\longrightarrow$ IRL)

	Sugar Task	Person Task
control	2.03	2.68
original	2.71	3.92
memory training	4.77	5.45
simple rule	5.32	5.55

Model Data (IDN+IRL $\longrightarrow$ RER)

	Sugar Task	Person Task
control	2.02	2.27
original	2.89	3.90
memory training	4.51	5.43
simple rule	5.05	5.37

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+RER $\longrightarrow$ IRL	0.247	0.445	0.048
IDN+IRL $\longrightarrow$ RER	0.237	0.306	0.167

Figure 7: The simulation of Stanley et al. (1989) with partial-full models. Each data point indicates the number of on-target responses per trial block.

bottom-up learning (as in RER).

When it comes to the question of which version of the model should be preferred, there is however a trade-off to be considered. The full model certainly produced the best fit compared with the other (partial) models, but it is also more complex than the other models. Comparing the full model with the partial models, the question is whether the increased complexity of the full model is worth the increase in accuracy. What we need here is a formal measure of complexity-accuracy trade-offs and a criterion for deciding when increased complexity is worth it. As there is a lack of formal measures of complexity-accuracy trade-offs currently, the answer to this question remains a matter of personal judgment at the present time.

Since there was no human learning curve available in this task (see Stanley et al 1989), there cannot be a comparison between human and model learning curves here. Therefore, learning curves are not presented.

We also tested a version of the model without RER rule learning and without full-blown Q-learning. As described below, the result was comparable, also capturing the essential characteristics of the human learning.

3.2 Stanley et al. (1989): Simulation 2

In this subsection, we present an alternative simulation of Stanley et al (1989). The reason for doing so is to demonstrate the *robustness* of the architecture (to a certain extent at least): Variations in representations and in details of learning algorithms do not alter the essential capability of the architecture to capture the human performance in this task.

Model Setup. In the bottom level, we used one IDN, with the simplified Q-learning algorithm. The network received feedback at each step, concerning whether the resulting system output was on target or not. Therefore, the simplified Q-learning algorithm applied easily to this setting. We used 168 input units, 40 hidden units, and 12 output units. There were 7 groups of input units, each for a particular (past) time step, constituting a moving time window. Each group of input units contained 24 units, in which half of them encoded 12 system output levels and the other half encoded 12 system input levels at a particular step. There were a total of 14 input dimensions. The 12 output units (constituting one output dimension) indicated the 12 levels of subjects' input to the system.

In the top level, IRL was used. The rule deletion threshold ($threshold4$) was set at 0.15 for simu-

Human Data

	Sugar Task	Person Task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Model Data

	Sugar Task	Person Task
control	2.276	2.610
original	2.952	4.187
memory training	4.089	5.425
simple rule	4.073	5.073

Mean-Squared Errors

	Total	Sugar Task	Person Task
IDN+IRL	0.187	0.134	0.240

Figure 8: The simulation of Stanley et al (1989) using alternative encodings and algorithms. Each data point indicates the number of on-target responses per trial block.

lating control subjects. To capture the verbalization condition, the rule deletion threshold was raised to 0.35 to encourage more rule learning activities. [9] To capture the explicit instruction conditions, various means were used. In the "memory training" condition, each of the 12 examples was wired up at the top level as rules (in the form of $P_1 \longrightarrow W$). In the "simple rule" condition, the rule (as described earlier) was wired up at the top level. The response was measured over a period of 20 blocks for each simulated subject. In addition, as before, for simulating the person task (a common, everyday task), we used pre-training before data collection, to capture the prior knowledge that subjects likely had in this type of task.

Simulation Results. As before, our simulation captured the verbalization effect in the human data. We used a t test to compare the "original" group with the control group in the model data, which showed a significant improvement of the original group over the control group ($p < .01$), the same as the human data. See Figure 8.

Our simulation also captured the explicit instruction effect, as shown in Figure 8. We used pair-

[9] Rules below that value were deleted as long as they had been applied at least 5 times.

wise t tests to compare the "memory training" and "simple rule" groups with the control group in the model data, which showed significant improvements of these two groups over the control group, respectively ($p < .01$).

Discussions. Like the simulation earlier, this simulation captured the verbalization effect and the explicit instruction effect. Both effects point to the positive role of the top level. They showed synergy between the top-level explicit processes and the bottom-level implicit processes. When the top level was enhanced, either through verbalization or through externally given explicit instructions, performance was improved, although such improvement was not universal (see Sun et al. 2001, 2005, Sun 2002).

This simulation, with alternative encoding of inputs and alternative learning algorithms (simplified Q-learning and IRL), was successful. Overall, although the earlier simulation captured the data somewhat better, this simulation nevertheless captured the data adequately. It shows the robustness of the CLARION architecture — Differences in model setups (e.g., Q-learning vs. simplified Q-learning, IRL vs. RER, 2-step time window vs. 7-step time window, etc.) did not significantly affect the outcome of simulation.

4 General Discussions

Theoretical Implications. Although there have been various controversies surrounding implicit learning, the existence of implicit processes in skill learning is not in question — What is in question is their extent and importance. We allow for the possibility that both types of processes (implicit and explicit) and both types of knowledge (implicit and explicit) coexist and interact with each other to shape learning and performance, so we go beyond the controversies and those studies that focused mostly on the minute details of implicit learning (Gibson et al 1997).

The incorporation of both processes allows us to ask the question of how synergy is generated between the two separate, interacting components of the mind (i.e., the two types of processes). The model may shed some light on this issue. Sun and Peterson (1998) did a thorough computational analysis of the source of the synergy between the two levels of CLARION in learning and in performance. The conclusion, based on the systematic analysis, was that the explanation of the synergy between the two levels rested on the following factors: (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 bottom-up rule learning criterion used in CLARION. For the sake of brevity, we will not repeat the analysis here. See Sun and Peterson (1998) for details. It is very likely, in view of the match between the model and the human data as detailed in this paper, that the corresponding synergy in human performance results also from these same factors in the main.

Clearly, there may be alternative interpretations of the data that we presented, that do not involve the assumption of two levels and may be more or less equally compelling. However, the two-level approach provides a consistent, theoretically motivated, and principled framework (Sun 1995, 2002). The approach succeeds in interpreting many findings in skill learning that have not yet been adequately captured in computational modeling (such as bottom-up learning and synergy effects; Sun et al 2001, 2005) and points to a way of incorporating such findings in a unified model that has significant theoretical implications.

Which Model is Superior? It was shown in the simulations that IRL contributed more to the capturing of the human data than RER. The simulations suggested that it might be the case that in human learning of this task, explicit hypothesis testing learning (as captured by IRL) was more important than implicit-to-explicit bottom-up learning (as in RER). However, there is clearly a trade-off to be considered: The full model certainly produced the best fit compared with the partial models, but it is also more complex than these models. The question is whether the increased complexity of the full model is worth the increase in accuracy. What we need here is a formal measure of complexity-accuracy trade-offs and a criterion for deciding when increased complexity is justified. Since there is a lack of formal measures of complexity-accuracy trade-offs, the answer to this question remains a matter of personal judgment.

The simulations reported here demonstrated the usefulness of explicit learning (IRL or RER) in capturing various effects in human skill acquisition (Sun et al 2001, 2005). Regardless of whether RER or IRL was involved, explicit rule learning captured explicit knowledge in human skill acquisition.

Our model (based on CLARION) is the most comprehensive model for process control tasks to date. Among other things, it includes all of the following: implicit learning, explicit hypothesis testing, and implicit-to-explicit extraction. It also includes both instance-based learning and rule-based learning in a unified manner. For instance, the RER learning algorithm starts with extracting concrete instances (from the bottom level) and can be either instance-based or rule-based, depending on generalization/specialization parameters within RER. The generalization/specialization parameters within RER may either discourage or encourage generalizing concrete instances (extracted from the

bottom level) into more abstract rules, depending on threshold values as described in Section 2. Moreover, our model includes both implicit and explicit learning (RER and IRL), and both bottom-up and top-down learning (potentially). Our model can also easily incorporate pre-existing heuristics as hypothesized in some theories (e.g., Dienes and Fahey 1996, Fum and Stocco 2003a,b).

Comparisons. As a result of its distinct emphasis, CLARION is clearly distinguishable from existing unified theories/architectures of cognition, such as SOAR, ACT, and EPIC. For example, SOAR (Rosenbloom et al 1993) is different from CLARION, because SOAR makes no clear distinction between explicit and implicit learning, and all learning is based on specialization, using only symbolic forms of knowledge. EPIC (Meyer and Kieras 1997) is also different from CLARION, because it makes no implicit/explicit distinction either, although it incorporates elaborate motor and perceptual processes as part of a cognitive theory/architecture.

Anderson (1983, 1993) and Anderson and Lebiere (1998) put forth two similar models: ACT* and ACT-R. 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. The ACT series of models is different from CLARION because traditionally it focuses mainly on top-down learning (from declarative to procedural knowledge). While ACT relies mostly on top-down learning, CLARION can proceed completely bottom-up (from procedural to declarative knowledge); therefore, it does not rely on existing, externally given, verbally imparted knowledge as much as ACT does. Note that CLARION can also proceed in a top-down direction (Sun 2002). Therefore, it accounts better for the interaction between different types of knowledge.

We may compare our model with other models dealing specifically with process control tasks. There have been in the past a number of simulation models for process control tasks, for example, Dienes and Fahey (1996), Gibson et al (1997), Lebiere, Wallach, and Taatgen (1998), Taatgen and Wallach (2002), Fum and Stocco (2003a,b), and so on. The key difference between our model and these other models is that our model is more complete and more comprehensive: It encompasses both instance-based and rule-based learning, among other things.

More specifically, Dienes and Fahey (1996) showed that, depending on task parameters, either instance-based or rule-based model might be more appropriate. The key determinant lies in the

salience of task stimuli. Instead of having two separate models as in Dienes and Fahey (1996), our model encompasses both instance-based and rule-based processes. Recall that RER learning starts with extracting concrete instances (from the bottom level) and can then become either instance-based or rule-based, depending on generalization/specialization parameters within RER, which may either discourage or encourage generalizing concrete instances (extracted from the bottom level) into more abstract rules. Therefore, our model includes the two models (rule-based and instance-based) of Dienes and Fahey (1996) as two special cases, both within the same generic model. Moreover, it is not clear how the models of Dienes and Fahey (1996) can account for gradual explication of implicit knowledge, for example, as reported in Stanley et al (1989), Willingham et al (1989), and Sun (2002).

Taatgen and Wallach (2002) showed that a model based on ACT-R could account for process control task data, using (essentially) the idea of instance-based learning from Dienes and Fahey (1996). Our comparison above between our model and the models of Dienes and Fahey (1996) applies to this model as well.

Fum and Stocco (2003a,b) showed that both Dienes and Fahey (1996) and Taatgen and Wallach (2002) failed to account for some human data concerning changing target levels. They argued that a set of simple heuristics competing to take control could best account for such data. While their results were highly interesting, it appears that both instance-based and rule-based processes are highly relevant to human performance in this task, as demonstrated by Stanley et al (1989), Dienes and Fahey (1996), Taatgen and Wallach (2002), Slusarz and Sun (2001), etc., in addition to generic demonstrations of the importance of instance-based and rule-based learning in the literature (e.g., Logan 1988, Sun 2002). In our own human experiments using process control tasks, we have also found numerous cases of concrete instances and abstract rules in verbal protocol data. Therefore, there appears to be a need to include instance-based and rule-based processes in models of process control tasks, such as done in CLARION.

Future Work. We would like to verify our computational analysis through further empirical studies of human performance in this task. Further experiments may verify (1) the relative contributions of different learning processes, and (2) the various effects that we have discussed so far. Let us discuss the two aspects one by one. In terms of verifying the relative contributions of different learning processes, we know of no existing methodology that can help us to tackle this issue. This aspect appears to be rather difficult, and thus remains a long-term goal of our research. In terms of verifying various effects of the implicit/explicit interaction, we have found many confirming results in the literature (Sun 2002, Sun et al 2005). For instance, in terms of the verbalization effect, as reviewed before, Stanley et al

(1989) and Berry (1983) found that under some circumstances concurrent verbalization could help to improve subjects' performance in a process control task. Reber and Allen (1978) similarly showed in artificial grammar learning that verbalization could help. Ahlum-Heath and DiVesta (1986) found that verbalization led to better performance in learning Tower of Hanoi. Likewise, in terms of explicit search instructions, as demonstrated by Berry and Broadbent (1986, 1988), Stanley et al (1989), and Reber et al (1980), they can facilitate or hamper task performance. For example, Reber et al (1980) found that, depending on the ways stimuli were presented, explicit search might help or hamper performance. Owen and Sweller (1985) and Schooler et al (1993) found that explicit search hindered learning. In terms of explicit how-to instructions, Stanley et al (1989) found that such instructions helped to improve performance significantly (although the findings in Berry and Broadbent (1988) were more ambivalent). Thus, this aspect is fairly well established. However, we may further explore this aspect in our future work.

5 Concluding Remarks

This work highlights the importance of the interaction of implicit and explicit processes in skill learning. Furthermore, it explores the relative contributions of implicit learning, explicit learning, and implicit-to-explicit bottom-up learning.

We capture such interactions through CLARION, which includes all types of processes (implicit or explicit). The computational modeling reveals something new in the existing data (cf. Gibson et al 1997, Lebiere et al 1998, Fum and Stocco 2003a,b). The contribution of this model lies in capturing human data in skill learning through the interaction of various processes, and also in verifying the computational feasibility and psychological plausibility of both bottom-up learning and independent hypothesis testing learning (Sun et al 2001, Sun 2002).

Acknowledgments

This work was carried out while the authors were supported in part by ARI grants DASW01-00-K-0012 and W74V8H-04-K-0002 (to Ron Sun and Bob Mathews).

References

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.

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

J. Anderson and C. Lebiere, (1998). *The Atomic Components of Thought*, Lawrence Erlbaum Associates, Mahwah, NJ.

D. Berry, (1983). Metacognitive experience and transfer of logical reasoning. *Quarterly Journal of Experimental Psychology*, 35A, 39-49.

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

S. Chaiken and Y. Trope (eds.), (1999). *Dual Process Theories in Social Psychology*. Guilford Press, New York.

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

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, (1994). Attention and awareness in sequence learning. *Proc. of Cognitive Science Society Annual Conference*, 330-335.

A. Cleeremans, (1997). Principles for implicit learning. In D. Berry (Ed.), *How Implicit Is Implicit Learning?* pp. 195-234. Oxford University Press, Oxford, UK.

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

Z. Dienes and R. Fahey, (1995). The role of specific instances in controlling a dynamic system. *Journal of Experimental Psychology: Learning, Memory and Cognition*. 21, 848-862.

D. Fum and A. Stocco, (2003 a). Instance vs. rule based learning in controlling a dynamic system. *Proceedings of the International Conference on Cognitive Modeling*, 105-110. Bamberg, Germany: Universitats-Verlag Bamberg.

D. Fum and A. Stocco, (2003 b). Outcome evaluation and procedural knowledge in implicit learning. *Proceedings of the 25th Annual Conference of the Cognitive Science Society*. Boston, MA: Cognitive Science Society, 426-431.

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

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.

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

S. Keele, R. Ivry, E. Hazeltine, U. Mayr, and H. Heuer, (1998). The cognitive and neural architecture of sequence representation. Technical report No.98-03, University of Oregon.

C. Lebiere, D. Wallach, and N. Taatgen, (1998). Implicit and explicit learning in ACT-R. *Proc. of ECCM'98*, pp.183-189. Nottingham University Press.

P. Lewicki, M. Czyzewska, and H. Hoffman, (1987). Unconscious acquisition of complex procedural knowledge. *Journal of Experimental Psychology: Learning, Memory and Cognition*. 13 (4), 523-530.

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.

T. Mitchell, (1998). *Machine Learning*. Addison-Wesley, Reading, MA. McGraw-Hill, New York.

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

E. Owen and J. Sweller, (1985). What do students learn while solving mathematics problems? *Journal of Experimental Psychology*, 77 (3), 272-284.

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

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.

D. Rumelhart, J. McClelland and the PDP Research Group, (1986). *Parallel Distributed Processing*. MIT Press, Cambridge, MA.

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

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

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

D. Shanks, (1993). Human instrumental learning: A critical review of data and theory. *British Journal of Psychology*. 84, 319-354.

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

P. Slusarz and R. Sun, (2001). The interaction of explicit and implicit learning: An integrated model. *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, Edinburgh, UK. pp.952-957. Lawrence Erlbaum Associates, Mahwah, NJ. 2001.

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

W. Stanley, R. Mathews, R. Buss, and S. Kotler-Cope, (1989). Insight without awareness. *Quarterly Journal of Experimental Psychology*. 41A (3), 553-577.

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, (1999). Accounting for the computational basis of consciousness: A connectionist approach. *Consciousness and Cognition*, Vol.8, 529-565.

R. Sun, (2002). *Duality of the Mind*. Lawrence Erlbaum Associates, Mahwah, NJ.

R. Sun, E. Merrill, and T. Peterson, (1998). A bottom-up model of skill learning. *Proc.of 20th Cognitive Science Society Conference*, pp.1037-1042, Lawrence Erlbaum Associates, Mahwah, NJ.

R. Sun, E. Merrill, and T. Peterson, (2001). From implicit skill to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*. Vol.25, No.2, pp.203-244. 2001.

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

R. Sun, P. Slusarz, and C. Terry, (2005). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review*, Vol.112, No.1, pp.159-192.

R. Sun and X. Zhang, (2002). Top-down versus bottom-up learning in skill acquisition. *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, Fairfax, VA. pp.861-866. Lawrence

Erlbaum Associates, Mahwah, NJ. 2002.

R. Sun and X. Zhang, (2003). Accessibility versus action-centeredness in the representation of cognitive skills. *Proceedings of the Fifth International Conference on Cognitive Modeling*, pp.195-200. Universitats-Verlag Bamberg, Bamberg, Germany.

N. Taatgen and D. Wallach, (2002). Whether skill acquisition is rule or instance based is determined by the structure of the task. *Cognitive Science Quarterly.*

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

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