COGNITIVE THEORIES OF SKILL ACQUISITION *

Michael E.J. MASSON
University of Victoria, Canada


Three theories of cognitive representation are described, with emphasis on their implications for issues related to motor skill acquisition. Production system models make a distinction between declarative and procedural knowledge, and skilled performance is assumed to be based on procedural knowledge that is not ordinarily verbalizable or available to consciousness. Neural network models rely on error detection and correction, in a manner reminiscent of closed-loop theory, to develop a distributed representation of knowledge that captures relationships between task components. Instance theories of skill acquisition are founded on the assumption that expert performance derives from automatic retrieval of memory for individual training episodes. The instance memory approach contrasts with the schema theory of skill learning. In general, the cognitive systems described here constitute forms of representation that typically are not open to modification by intentional processes such as mental practice. They are constructed and influenced instead by direct experience with a task.

The interest in cognitive representational systems as a basis for motor skill learning, exemplified by Adams (this issue), is an encouraging sign for cognitive psychologists. Application of cognitive theory to other domains offers an excellent opportunity to test the generality of propositions intended to capture fundamental aspects of cognitive architecture. My objective is to encourage further attempts to extend cognitive theories of skill acquisition to the motor learning domain by describing a number of recent theoretical developments that complement those presented by Adams.

* Preparation of this article was supported by a grant from the Natural Sciences and Engineering Research Council of Canada. The article is based on an invited presentation given at the annual meeting of the Canadian Society for Psychomotor Learning and Sport Psychology in Victoria, British Columbia, October 20, 1989. I am grateful to Timothy Lee and Geraldine van Gyn for extending the invitation and for valuable suggestions that aided in the preparation of this article.

Correspondence should be addressed to M.E.J. Masson, Dept. of Psychology, University of Victoria, Victoria, British Columbia, Canada V8W 2Y2.
In his review, Adams advocated three concepts, engrained in cognitive theories of learning, as potential tools for constructing a theory of motor learning. The centrally important concept in Adams' scheme is mental imagery as a representational system (Paivio 1971). This system has been strongly implicated in the two other concepts discussed by Adams, observational learning and mental practice. A critical contribution of mental imagery is to the learning and retention of images capable of generating long movement sequences. The availability of this representational system is central to the claim that motor skill can be improved even in the absence of actual physical movement. Critical movements may simply be observed or imagined, leading to the development of imaginal representations that can significantly contribute to skilled performance. In this view, mental imagery is a cognitive representational system that may be directly linked to motor control processes and sufficiently fine-grained to capture the subtleties of movement skill (Adams 1986).

An important characteristic of the hypothesized role for mental imagery in skilled motor performance is that nonverbal images may be joined with verbal descriptions of movement requirements to regulate the actual execution of motor behavior. This view places the development and application of mental images in the realm of conscious control, in the sense that the relevant representations may be introspected and verbally described. A portrayal of this sort seems incomplete in light of the observation that highly skilled performance is often associated with the withdrawal of attention and the inability to provide a verbal description of one's performance (Shiffrin and Schneider 1977). A more compelling account of skilled performance seems to require the incorporation of representational systems that are geared toward operating outside the bounds of conscious awareness. Three such systems, all of which are currently at the leading edge of research and theory building in cognitive psychology, are summarized here and offered as potential frameworks for constructing more comprehensive theories of motor skill development.

**Production systems**

A critical factor in the development of cognitive theories of knowledge representation and application has been the goal-directed nature
Table 1
A sample production for word substitution in a text editor.

<table>
<thead>
<tr>
<th>IF</th>
<th>the goal is to replace WORD1 with WORD2</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEN</td>
<td>set as subgoals</td>
</tr>
<tr>
<td></td>
<td>(1) to find WORD1</td>
</tr>
<tr>
<td></td>
<td>(2) then to delete WORD1</td>
</tr>
<tr>
<td></td>
<td>(3) then to insert WORD2</td>
</tr>
</tbody>
</table>

of intellectual behavior. This aspect of performance has been effectively expressed in computer simulation models known as production systems (Anderson 1983; Card et al. 1983; Newell and Simon 1972). These systems are collections of productions that consist of condition–action pairs and are responsible for initiating appropriate actions under specified conditions. A production typically is expressed in the form of an IF–THEN statement with execution of the specified action contingent upon instantiation of the stated condition. For example, a production system model of knowledge about using a computer text editor might include a production for replacing one word with another, as shown in table 1. Notice that in this example, both the condition and the action involve coordination of goals. A complete production system would consist not only of productions that result in the instantiation of goals, but also of productions that execute particular physical actions when relevant goals and environmental conditions are established.

In the context of cognitive representation of knowledge underlying skilled performance, examination of the production system model developed by Anderson (1983, 1987) is especially enlightening. In fact, I would argue that this system, named ACT*, offers a promising answer to one of the key questions raised by Adams (this issue): How is a cognitive representation translated into action?

In the ACT* model a clear distinction is made between two kinds of knowledge. Declarative knowledge consists of information that can be described verbally, retained and manipulated in working memory, and is represented in the form of propositions or mental images. For example, a verbal description of how to carry out an action, such as shifting an automobile’s manual transmission, could be coded as a set of propositions and even mentally rehearsed in an effort to improve performance. Procedural knowledge is embodied as production systems with condition–action pairs represented in long-term memory.
Unlike declarative knowledge, procedural knowledge is not verbalizable or open to conscious introspection. For a given individual, different aspects of expertise in a specific domain are assumed to coexist in these two different representational formats. In the ACT* system, however, procedural knowledge governs skilled performance.

An important aspect of the ACT* model is its characterization of the movement from novice to expert performance as a transition from control by declarative knowledge to control by procedural knowledge. Anderson (1983) refers to this process as *proceduralization*. Initially, a novice is assumed to represent knowledge about a task in a declarative form such as verbal propositions (e.g., Kintsch 1974). At this stage performance requires maintaining in working memory the declarative representation of task components and their relationships. With practice at the task a production system develops that is eventually able to generate skilled performance. The productions represent task-specific procedures in long-term memory and are activated without requiring knowledge about the procedures to be retrieved into working memory. The characterization of skill development as a movement from emphasis on declarative to procedural knowledge is consistent with views of motor skill development in which a learner makes transitions through cognitive, associative, and autonomous stages (Fitts 1964), or from a verbal-motor to a motor stage (Adams 1971).

In the ACT* theory proceduralization consists of constructing domain-specific productions from relevant declarative knowledge and generic productions. An example of each of these three knowledge components for the task of shifting a manual transmission into reverse is shown in table 2. The generic production acts as a template and declarative knowledge of the task is used to fill out the template to produce a domain-specific production. The components shown in table 2, of course, do not constitute an exhaustive list of the elements necessary for this task.

Once a set of productions has been developed, task performance can go forward without reinstating in working memory the domain-specific declarative information that gave rise to the productions. A possible consequence of proceduralization of task performance, then, is that the declarative representation may be forgotten or become very difficult to reinstate. A compelling example of this phenomenon involves frequently dialed telephone numbers (Anderson 1976). With frequent repetition a specialized procedure for dialing a number may be devel-
Table 2
Example of knowledge components involved in proceduralization.

(a) Declarative knowledge
   To shift into reverse, disengage clutch, move gear to upper left, etc.

(b) Generic production
   IF the goal is to achieve state $X$
       and $M$ is a method for achieving $X$
   THEN set as a subgoal to apply $M$

(c) Domain-specific production
   IF the goal is to go into reverse
   THEN set as subgoals
       (1) to disengage the clutch
       (2) then to move the gear to upper left
       (3) then to engage the clutch
       (4) then to push down on the gas

oped and used without requiring a declarative representation of the number to be retrieved into working memory. Continued use of the procedural representation eventually makes it very difficult to retrieve the unused declarative representation so that the only way one can remember the number is through execution of the procedure for dialing it.

Verification of the ACT* theory of skill acquisition depends on securing a means of distinguishing cases in which task performance is governed by procedural as opposed to declarative knowledge. Two empirical approaches to this issue are described here. First, proceduralization of a skill implies disappearance of the ability to provide a complete verbal account of performance and, importantly, an end to the reliance on working memory representation of declarative knowledge while executing the task. Anderson (1987) described an unpublished study involving a dual-task paradigm that provides evidence for this prediction. The primary task consisted of working on a visual text editor and the secondary task required memorization of facts presented in the auditory modality. In one condition the facts were based on the rules for operating a different text editor and in the other condition the facts were irrelevant to text editors.

In the early stages of learning the text editor, fact memory scores in the first condition were poor, but they improved with practice. Apparently, proceduralization reduced dependence on working memory representations of declarative knowledge about the editor being learned
and interference with attempts to memorize conflicting facts was reduced. In contrast, fact memory scores in the other condition were higher than in the first condition at the outset and did not improve with practice. We can conclude that the irrelevant facts did not suffer direct interference from declarative knowledge used in the early stages of learning, and so memory for them did not improve as the source of potential interference faded.

Allard and Burnett (1985) described a similar demonstration of the movement away from reliance on working memory representation of declarative knowledge. Softball batters of low or high skill attempted to hit pitches while engaged in a secondary task. Players at the higher skill level might be expected to have reached the stage of proceduralization of batting performance. For players at the lower skill level batting performance might still require working memory representation of declarative knowledge. Consistent with this idea, the highly skilled players' ability to hit was not adversely affected by an auditory digit search task, although the less skilled hitters' performance suffered. On the other hand, when the secondary task involved a visual memory task that required attention to be devoted to irrelevant parts of the pitcher's body, performance of the high skill, but not the low skill, players suffered, indicating that the proceduralized skill depended on information concerning critical features of the pitcher's movements.

A second approach to the issue of demonstrating proceduralization of task performance is based on transfer of training. An important prediction from the ACT* model is that skill transfer from one task to another depends on the number of productions shared by the production systems that govern performance on the two tasks (Anderson 1987). This view is a variant of Thorndike's (1903) identical elements theory of transfer, and it predicts no negative transfer for proceduralized tasks. That is, performance of one task does not become less effective because another task is learned. In the worst case, two tasks will share no productions and will exhibit no transfer. This characteristic of procedural knowledge contrasts sharply with what is known about declarative knowledge. The classic verbal learning literature clearly shows, for example, that negative transfer is common in paired associate learning paradigms. Negative transfer is particularly strong in the AB–ABr design where subjects first learn a list of paired associates, then learn a new list consisting of the same stimulus and response items, but with the pairs rearranged (Postman 1971).
Singley and Anderson (1985) devised a procedural knowledge variant of the AB–ABr paradigm using the EMACS text editor and a variant of it in which the functionality was the same but the assignment of keystroke sequences to functions was rearranged. For example, in EMACS the sequence CONTROL-N moves the cursor down one line; in the variant CONTROL-D might be assigned this function whereas in EMACS CONTROL-D deletes a character. The control group of subjects was trained on the EMACS editor for six days. The experimental group followed a regimen of two days with EMACS, two days with the variant, and two more days with EMACS. Performance was measured in terms of time required to carry out an edit function. If negative transfer were to occur, it would be expected on the third day when the experimental group was transferred to the variant.

Although the experimental subjects took slightly longer to perform edits on the third day (about 50 s) than on the second day of practice (about 44 s), they were much faster relative to their first day of practice (about 90 s). This result means that there actually was positive transfer from practice on EMACS to learning the variant editor. Moreover, when the experimental group returned to the EMACS editor on the last two days (their third and fourth days of practice on EMACS), their performance was virtually equal to that of the control subjects on their third and fourth days of practice with EMACS. The two-day foray with the variant of EMACS did not move the experimental subjects off the learning curve for the original EMACS.

The positive transfer and ability to maintain the original learning curve both appear to be the result of an important role played by productions relevant to the goal structure that underlies proceduralized skill on the EMACS text editor and its variant. An example of a production of this type is shown in table 1. That production specifies the subgoals associated with exchanging one word for another and would be relevant in either version of EMACS. Learning goal-relevant productions in EMACS provided an important foundation for learning the variant editor as well, despite the reassignment of keystroke sequences. The cost of learning the new assignments was reflected in the fact that experimental subjects were slower on day three (when they first tried the variant) than on day two. The set of productions common to both versions of EMACS, however, received additional strengthening during the two days of practice with the EMACS variant, enabling the EMACS learning curve to show continuous improvement when experimental subjects returned to that editor.
The ACT* theory and the empirical work offered to support it have established the existence of significant differences between declarative and procedural knowledge. The successful development of production system models that capture procedural knowledge and simulate skilled task performance (e.g., Anderson 1987; Kieras and Bovair 1986) illustrates the powerful role of procedural knowledge in governing expert performance. Increased reliance on the procedural knowledge system at advanced levels of skill appears to set serious limitations on the potential contribution to skilled performance that might be made by mental practice, observation, and other activities grounded in declarative knowledge.

Neural network models

A second form of representational system that holds significant promise as a medium for the exploration of motor skill learning emphasizes the distributed nature of mental representation. Neural network models are styled after the concept of densely interconnected cells that characterizes the organization of the brain. Although these models usually are not intended as direct analogues of brain function and structure, they are meant to capture some of the critical features of brain systems. Neural network models address a number of issues that are central to the field of motor learning, particularly the role of knowledge of results, error detection, and error correction.

The critical idea regarding representation in neural network models is that knowledge consists of weighted connections among an entire network of processing units (J.A. Anderson 1977; Rumelhart and McClelland 1986b). In many applications of neural network models the type of knowledge being simulated involves relationships between patterns such as the transformation of verbs from present to past tense, or the coordination of limb movement to point at an object fixated by the eye. A pattern, such as a verb or an eye position, typically is represented as a specific pattern of activation across a subset of the processing units in the system. Some units in the subset are highly activated while others are at low levels of activation. The system expresses its knowledge by propagating and transforming the pattern of activation in the first subset of units (e.g., an eye position), via weighted connections, into a pattern of activation across a different
subset of units that constitutes a response to the initial pattern (e.g., a pointing movement). An example of how this kind of system can be applied to the exploration of movement dynamics was provided by Bullock and Grossberg (1988), who developed a neural network model that accounts for the dynamics of arm movements and the formation of the trajectory for such movements.

To illustrate the general approach of neural network models, I have chosen one type of model to explore in limited detail. My goal is to demonstrate how this type of model acquires, represents, and uses knowledge to perform a task. This exercise serves to highlight the potential relevance of neural network models to aspects of motor learning theory.

A diagram illustrating the structure of a neural model is shown in fig. 1. There are three sets or layers of processing units, with units at each level connected to every unit at the adjacent level(s). The system is designed to generate a response, expressed as a pattern of activation across the units in the output layer, to a stimulus that consists of a pattern of activation fed into the input layer. The system's response to a stimulus pattern is determined by that pattern's profile of activation across the input units and by the strengths of each of the connections in the system. Activation propagates from the input units through the internal representation or hidden layer, then on to the output layer. At each transition between layers, the activation received by a unit in the receiving layer (e.g., hidden units receive activation from input units) is computed as the sum of weighted activation sent by units in the
immediately preceding layer. For example, the activation received by a hidden unit would be expressed by the following equation:

\[ a_i = \sum_j w_{ij} o_j. \]

In this equation, \( a_i \) refers to the activation received by hidden unit \( i \), \( o_j \) represents the activation transmitted from input unit \( j \), and \( w_{ij} \) is the strength or weight of the connection between the two units. Connection weights may be positive or negative, the latter capturing the idea of inhibitory connections between units. Notice that summation runs across all units in the input layer.

Once the amount of activation coming into a unit has been computed it is transformed by some nonlinear function (e.g., a sigmoid function) into a value that represents the current activation level of the unit. The nonlinear function is used to constrain the range of activation values that a unit can take on, usually between 0 and 1. The activation in the hidden units is then propagated to the output units using the same equation, and the nonlinear function is used to compute the pattern of activation across the output layer.

The critical information in this system is the pattern of connection weights among units. These connection weights are built up through a series of training trials in which the system is exposed to pairs of input–output patterns. The usual approach to training a neural network system is to initially set the connection weights to small random values. In the early training trials, then, the system will generate inappropriate output patterns when input patterns are presented. A learning procedure referred to as back propagation is used to make adjustments to connection weights in an effort to make the output patterns produced by the network correspond more closely to the desired output patterns. The procedure involves comparing the observed output pattern to the desired pattern and computing the amount of deviation or error in each unit. The size of this error is then used to compute the degree to which the connection weights attached to the unit must be changed. This process is a form of blame assignment, in which connections with the greatest weight will be changed the most when an error is observed, and the amount of change will be larger when the observed error is larger. Information about size of error is also propagated to the hidden units so that changes are also applied to
the weights of the connections between the hidden and input units. A full explanation of the back propagation procedure can be found in Rumelhart et al. (1986).

Through a sometimes long series of training trials the system evolves a set of connection weights that enable it to produce a relatively accurate pattern of activation in the output units in response to any of the stimuli used in training. Moreover, assuming that some coherent relationship exists between the stimulus and response patterns, the system will also show rather strong transfer when presented with novel stimuli. For the type of network described here, the ability to represent knowledge about input–output patterns used in training and the capacity for generalizing from training instances lies in the use of an internal representation layer of units. Early neural network systems, called perceptrons, had only an input and an output layer and were not capable of learning sets of input–output patterns that constituted complex relationships (Minsky and Papert 1969).

The characteristics of the class of neural network models I have described are relevant to a number of issues in the motor learning literature. For example, it is clear that learning in the system depends on knowledge of performance or information feedback, expressed in terms of the deviation between the actual output of the system and the desired output. In addition, neural network models offer an explicit theory of error handling and correction in the form of the back propagation procedure.

To demonstrate the heuristic value of neural network modelling, I applied a simple version of the type of network described here to the problem of simulating a set of empirical results in the motor learning literature. The example I selected was the contextual interference effect. In this paradigm a number of variants of a task are learned, with trials on each variant presented either in blocked or random order. The blocked order of presentation results in superior training performance, but the surprising result is that using a random order during training yields superior retention (Lee and Magi11 1983; Shea and Morgan 1979).

The contextual interference effect can be explained at an intuitive level using the neural network model. In the model knowledge about how to perform a variant of a task can be represented as pairs of input–output patterns. When training involves blocked presentation, the system works with a relatively small set of training patterns in each
block and should therefore develop an appropriate set of connection weights rather quickly, making for rapid learning. When the system moves to a new block the weights must be modified, but because the task variants (and the sets of input–output patterns representing the variants) are related the modifications are not radical. The pattern of weights learned in the prior block should be similar to the pattern that must be developed in the current block. In the random condition, however, the system is faced with the entire collection of input–output patterns throughout training and it must develop a set of weights capable of simultaneously representing this larger set. Given the greater number and the variety of patterns that must be mastered, the random condition should produce slower learning. But during a retention test in which each variant is tested in turn, blocked training should be at a disadvantage. The problem for the blocked condition is that the set of weights established by the end of training is most appropriate only for the most recently learned task variant. On the other hand, when the system is trained in the random condition it is prepared to handle any variant used during training. In the random condition the weights are developed to capture the entire collection of input–output patterns.

To simulate these results a computer program was used to construct a network with 6 input, 12 hidden, and 4 output units. In the program the units consisted of variables that could take on any real value. The weights of the connections between the units also consisted of real valued variables in the program. Two task variants were simulated by creating two sets of related input–output patterns, as shown in table 3. Each pattern was a set of 0s and 1s representing the level of activation in each input or output unit. The four patterns associated with each

<table>
<thead>
<tr>
<th>Task variant</th>
<th>Input patterns</th>
<th>Output patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 0 1 0</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1 1 1</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td></td>
<td>1 0 1 0 0 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0 1 0 0 0 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>2</td>
<td>1 1 1 1 0 1</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td></td>
<td>1 0 1 0 1 0</td>
<td>1 0 0 1</td>
</tr>
<tr>
<td></td>
<td>0 1 0 1 1 1</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1 0 1</td>
<td>0 0 1 0</td>
</tr>
</tbody>
</table>
task variant were taken as components that would be performed in sequence. To simulate training, the four input–output patterns of one variant were presented in a fixed order to the system. Each presentation of a pattern involved assignment to the input units of the string of 0s and 1s constituting the input pattern. Then the activation in the input units was propagated to the hidden units and finally hidden unit activation was computed and propagated to the output units. The resulting activation in the output units was compared to the specified output pattern (shown in table 3) to determine the amount of error in each output unit. The amount of error was then used to adjust the weights connecting the units.

In the simulation of the random training condition the four patterns representing one task variant were presented to the system in a fixed order. Then the four patterns representing the other variant were presented in a fixed order. The training regimen continued in this alternating fashion until each set of four patterns had been presented 15 times. The blocked condition was simulated in the same way except that the patterns for one variant were presented in a fixed order 15 times, then the patterns for the other variant were presented 15 times. For the transfer test, the two sets of four patterns were presented alternately four times each. The simulation was run twice in each condition, counterbalancing the order of presentation of the two variants of the task.

The simulated data consisted of the amount of error in the pattern of activation across the output units on each trial. The mean absolute error, averaged across the four output units, is shown in fig. 2. The pattern of error values followed the empirical data in all critical respects. The improvement in performance during training was greater for the blocked condition, but performance during the retention test was better in the random condition. The rationale for the model’s result given earlier is compatible with Lee and Magill’s (1983) explanation of the phenomenon. They claimed that random ordering of task variants required a solution to a task to be regenerated on each occasion. In the blocked condition, however, the solution could be ‘passively remembered’. From the model’s perspective, the correction of connection weights on each presentation in the random condition ensures that each variant continues to influence the evolving set of weights throughout training.

Neural network models are capable of learning complex sets of
input–output patterns, including those in which some logical rule governs the transformation of input to output patterns. An important characteristic of these models, however, is that they are able to do so without developing an explicit representation of the rule. Knowledge of the relationship between input–output patterns is captured within the connection weights. For example, Rumelhart and McClelland (1986a) developed a model that accurately simulates the pattern of errors made by children as they learn to transform verbs from present to past tense. The model reflects major features of this learning, including (a) early success with frequently occurring, irregular verbs (e.g., go + went), (b) later overregularization (e.g., go + goed), and (c) eventual success with both irregular and regular verbs. The model simultaneously learns about the transformation of regular verbs and about specific exception words. But it does not contain an explicit representation of the general transformation rule for verb tense. An important implication of the implicit representation of a rule system is that behavior may be governed by the system without explicit or conscious application of rules.
**Instance theory**

The representation of training experiences that constitute the basis of a skill has been a central issue in the construction of theories of skill acquisition. An influential view is that individual episodes are collectively represented in some form of schema that captures the essential or defining features of the skill (Schmidt 1975). Specific applications of skilled behavior are generated from the appropriate schema. Recent developments in cognitive psychology, however, have raised the possibility that memory for specific episodes or instances of experience may play a much stronger role in skill development than previously believed.

Logan (1988) has presented a theory of skill acquisition and automatization that emphasizes the accumulation of memory for instances. In Logan’s view, the movement from attention-demanding, effortful performance, to fluent, automatic execution consists of a transition from algorithm-based to memory-based performance. During the early stages of skill acquisition performance depends on the development and deliberate implementation of an algorithm. This level of skill corresponds to the reliance on working memory representations of declarative information discussed by Anderson (1987). The application of an algorithm, however, is slow and cumbersome because it depends on the conscious manipulation of information in working memory.

For example, learning to spell words may begin with application of a set of rules for generating a letter string from a word’s phonemic pattern and rules for exceptions (e.g., *i* before *e* except after *c*, etc.). The critical contribution of practice to skill development is the creation of memory for each practice instance. It is assumed that an episodic memory of each training instance is formed as an automatic consequence of experience. When a previously encountered item is repeated in the same task, the algorithm may be reapplied, but also memories for prior instances (e.g., occasions on which a word was spelled) will be automatically retrieved in parallel from long-term memory. Performance on the task may then be based either on completion of the consciously controlled algorithm or on a spelling automatically retrieved from memory. Extended practice produces a large collection of instances. When a highly practiced item is tested, it is very likely that one of these instances will be retrieved before the algorithm is completed, allowing a fast response to be based on memory retrieval alone.
These situations give rise to the impression that a task is performed automatically, or without attention.

Logan (1988) presented an analysis of hypothetical distributions of retrieval times for independently stored and simultaneously retrieved memories for instances and distributions of completion times for algorithms. By assuming that (a) execution of an algorithm and retrieval of instances from memory proceed in parallel and independently, and (b) performance depends on the first process that finishes, Logan was able to account for the ubiquitous power-function speed-up in task performance and the power-function reduction in variability of completion times that develop with practice. The reason for faster performance with more practice is that more instances involving an item are added to memory. When an item is repeated each prior instance is independently retrieved and the probability that at least one will be retrieved before the algorithm completes its run increases with the number of encoded instances. There are diminishing returns, however, and the average amount of reduction in response time with added instances progressively declines, yielding the power-function change in mean and variability.

The proposal that skill development is influenced by memory for specific instances is compatible with aspects of Adams' (1971) closed-loop theory, and particularly his concept of the perceptual truce. For Adams, the perceptual trace is a representation of a previous movement and its response-produced feedback. Over a series of trials a distribution of perceptual traces is assumed to form and the mode of this distribution determines the movement that is produced. This view is similar to Logan's (1988) instance theory in the sense that representations of previous training instances are preserved. But in closed-loop theory only the modal value of a distribution of traces influences performance, whereas instance theory assumes that all traces compete on an equal basis with the first one to be retrieved determining the response.

The instance view of skill acquisition entails a number of implications for the structure of training instances and subsequent patterns of transfer of training. In contrast to views such as schema theory, it is argued that skill is based on individual instances, rather than some form of general or schematic skill. In fact, the instance theory proposed by Logan suggests that the generic version of performance control occurs in the form of a slowly executed algorithm that eventually is
abandoned with practice. The practiced version of a skill should be highly specific to training experiences and may not be available under appropriately chosen circumstances.¹

These principles are illustrated by a study that examined how people learn to read typographically transformed words (Masson 1986). The task was to identify words printed in a mirror image font, with individual letters mirror-reversed but arranged in the usual left to right order to form words. One view of how people learn to identify words in this kind of font is based on the claim that some general transformational rule (e.g., mental rotation) is developed and applied to each word. From an instance-based view of skill acquisition, however, readers are learning new sets of visual patterns and the nature of the skill depends on the training instances used. To test the instance-based view, subjects were given training with words comprised of one half of the letters of the alphabet. The other letters of the alphabet were not presented during training. In a subsequent test phase subjects were presented with (a) words from the training phase, (b) new words consisting of letters that had appeared in words used in the training phase, and (c) new words consisting only of letters that had not appeared during training. Transfer of training was measured in terms of identification time, relative to performance in the training phase. As predicted from the instance-based view of skill acquisition, there was no evidence of transfer among items formed from untrained letters, although the other two conditions, particularly the condition involving training words, showed significant transfer effects. In the absence of any general skill, the lack of experience with half of the letters of the alphabet meant that there was no basis for improved performance on words consisting of those letters.

Conclusion

In conjunction with Adams (this issue), I have tried to advance the view that cognitive theories of knowledge representation offer a rich source of alternative ways of thinking about skill acquisition and

¹ Although I have chosen to present an extreme version of instance theory (Logan 1988), it should be noted that other frameworks exist in which memory for instances is assumed to coexist with schematic representations (e.g., Malt 1989).
transfer. Although these theories typically have been developed in the context of intellectual and perceptual tasks, I see no reason to expect the theories to fail catastrophically when applied to the psychomotor learning domain. In fact, many of the applications of these theories already border on or directly involve simulation of movement tasks (e.g., Bullock and Grossberg 1988).

In the collection of three approaches sketched here, we have seen a number of important links to classic theories of motor skill learning. For example, the detection and correction of error that is so critical in closed-loop theory (Adams 1971) is also central to neural network models of learning. The role of knowledge of performance in the learning exhibited by these systems offers a test bed for exploring ideas concerning schedules of feedback. The highly specific nature of transfer of training predicted by the instance-based view of skill acquisition and confirmed by empirical studies (Logan 1988; Masson 1986) suggests inadequacies in schema theories of skill (Schmidt 1975). The distinction between declarative and procedural knowledge that is so important in Anderson's (1983, 1987) ACT* theory provides a compelling account of differences in transfer of training that distinguish declarative and procedural knowledge representations.

Although Adams (this issue) has chosen to emphasize theories of representation that highlight the use of imagery and consciously controlled mental practice, I have attempted to make a convincing case in favor of representational systems that lie below the surface of verbalizability and imagery – systems that may operate for the most part outside the limits of awareness and intention. The systems I have discussed are no less cognitive, but they suggest further ways in which mental representation can influence the development and application of skilled performance. Like Adams, I am not yet ready to assert the bounds of cognitive representational systems.

References