Nerb, J., Ritter, F. E., & Langley, P. (2007). Rules of order: Process models of human learning. In F. E. Ritter, J. Nerb, T. O'Shea, & E. Lehtinen (Eds.), In order to learn: How the sequences of topics affect learning. 57-69. Oxford University Press.

Chapter 5. Rules of order: Process models of human learning

Josef Nerb and Frank E. Ritter and Pat Langley

March 23, 2006

Abstract

To fully understand sequential effects on learning in humans, we need a comprehensive theory of cognition. Such a theory should be complete enough to perform the task of interest and to learn like humans while doing so. These theories, called process models or cognitive models, can be broken down into aspects that do not change between tasks — the architecture — and aspects that do change — the knowledge. Where such models have been used to understand sequence effects on learning, they have proven to be very powerful. As an example, we present a model of behavior on a simple task that shows how an appropriate order can lead to significantly (16%) faster learning. However, despite their power, process models remain difficult to apply routinely. In response, we also discuss an alternative approach — abstract models — that may be more appropriate in some contexts.

1 Introduction

Science is concerned not only with data, as we discussed in the previous chapter, but with models or theories that explain those data. Because human cognition is dynamic and involves change over time, accounts of cognition often take the form of *process models*, which are sometimes also called *cognitive models*. In this chapter we review the form such models have taken and their relation to order effects in learning.

We begin by discussing the connection between artificial intelligence (AI) systems (e.g., as reviewed by Cornuéjols, Chapter 3), including those from machine learning and computational models of human behavior, including some illustrations of the latter. After this, we present a computational model of order effects on a cognitive task, cast within a particular but simplified theoretical framework. Next, we explore more broadly the possible sources of order effects within such models, and then briefly consider an alternative approach that models human behavior at a more abstract level. We close with some open problems in the area of modeling order effects and a charge to new modellers.

2 Process models in cognitive science

Many sciences use process models to explain the behavior of complex, dynamic systems. For example, physics often uses the formalism of differential equations to describe the relationships among quantitative variables (say, heat and temperature in a furnace) over time. Process models of human behavior have somewhat different requirements, as what changes over time are not primarily continuous variables, but rather qualitative structures in short-term and long-term memory as well as in motivational state.

Fortunately, computer languages provide formalisms that can be used to model the symbolic aspects of human cognition in the same way that differential equations are used in physics. Moreover, the field of AI has developed a variety of representations, performance mechanisms, and learning methods that can operate on many of the tasks that confront humans. Some AI work has little relevance to cognitive science because it makes no effort to constrain its methods to match psychological phenomena, but other AI systems have been developed with this goal explicitly in mind, and one can use them as computational models of human behavior. Indeed, some of the earliest AI systems, including EPAM (Feigenbaum & Simon, 1984) and GPS (Newell, Shaw & Simon, 1962) fall into this category.

2.1 The advantages of formal models

The advantage of formal models over informal ones is the same in cognitive science as in any other field. Rather than being sufficiently vague to handle almost any empirical result, detailed models of the processes that generate behavior lead to specific predictions that can be shown to be incorrect.[1]. Such results can thus lead to improved models that account for problematic findings and that make new predictions, leading to iterative progress towards more complete theories. Formal models also let us examine the internal states and mechanisms in the model that gave rise to the observed behavior, such as order effects. This lets us predict the effects of different conditions, such as alternative orders, without running subjects.

Another advantage of having a model's detailed behavior at hand is that it assists in analysing a subject's behavior. In particular, it lets us partition behavior into portions that the model can explain and those that it cannot, thus identifying anomalous observations. This in turn helps indicate where the model is incorrect and suggests where to improve it. Finally, a well developed, parameterized model can be used to classify subjects by their characteristics (e.g., Daily, Lovett & Reder, 2001), providing an account of individual differences.

2.2 Types of process models

Before examining how computational models can explain order effects observed in human learning, we must briefly review the major types. Early process models were compared to psychological data, but they incorporated rather idiosyncratic mechanisms and structures that were rarely shared within the research community. Over the past decade,

[1]Moreover, AI models actually carry out the tasks they address, which opens many possibilities for applications(e.g., Anderson & Gluck, 2001; Ritter et al., 2003).

researchers have imposed an increasing number of theoretical constraints across sets of models. These often take the form of *cognitive architectures* (Newell, 1990), which posit the structures and mechanisms of the cognitive system that are common across tasks. A cognitive architecture provides a framework for describing the sources of thought and behavior, including order effects.

Although there exist numerous computational models for aspects of cognition like categorization and long-term memory retrieval, we will focus here on frameworks for modeling more complex sequential tasks like problem solving and natural language. One widespread class of architectures used for such tasks is known as *production systems* (Jones, Ritter & Wood, 2000; Neches, Langley & Klahr, 1987; Young, 1979). This framework includes a long-term memory that contains productions or condition-action rules, which changes only slowly with learning, and short-term or working memory, which is far more dynamic. What varies across models in this framework is the contents of long-term memory.

A production system operates in cycles, on each step matching its rules against the contents of short-term memory (which may include representations of the environment), selecting one or more rules to apply, using their actions to alter short-term memory or the environment, and then iterating. Various learning methods exist for combining, generalizing, specializing, or otherwise modifying the rules in long-term memory. For example, a common approach involves making a larger rule out of several smaller rules that applied in sequence. This new rule can reduce the load on working memory and increase the speed of processing. If the choice of which smaller rules to apply was learned from extensive problem solving, the new rule can also constitute new knowledge about how to constrain future behavior. Examples of production systems that learn include Pavlik's ACT-R model (Chapter 10) and Ohlsson's HS system (Chapter 11).

A second well-studied framework for modeling sequential behavior is *recurrent neural networks* (for an introduction see Bechtel & Abrahamsen, 2002). Such models include a long-term memory composed of nodes, directed links connecting nodes, and weights on the links, with short-term memory consisting of temporary activations on the nodes. What varies across tasks are the number and connectivity of the nodes and the weights on links. Lane (Chapter 5) reviews this approach in more detail. A recurrent network also operates in cycles, on each cycle using the activations on 'input nodes' (at the lowest level) and the weight of links to compute the activations of higher-level nodes, ultimately calculating activation levels for output nodes that determine actions. The activations for some higher level nodes then replace those for the input nodes, and the next cycle begins. Learning typically occurs by propagating errors (differences between the desired and output values) downward through the network and modifying the weights to reduce these errors in the future.

Although production systems and recurrent neural networks are not the only classes of models used to model sequential human behavior, they are certainly the most widely used. Other architectural frameworks, including case-based and probabilistic ones, differ in their assumptions about representation, knowledge retrieval, and learning, but can be described using the terms of cognitive architecture and knowledge. Indeed, the most remarkable aspect of these architectures is not their differences but their similarities. All operate in cycles, in some sense matching against the contents of memory, taking actions to alter it, and then repeating the process. Their learning methods, despite their differences in operation, also tend to give similar effects, such as mastering simpler structures before more complex ones. We will return to this observation later in the chapter.

3 A simple model of order effects

An example model that produces an order effect in learning will clarify how one can use a cognitive architecture to understand human behavior and ground the rest of the discussion in more concrete terms. We first present the model's structure and mechanisms, after which we explain its behavior on a simple task that shows how order effects can arise.

We will assume that the example model is based upon a simple architecture that incorporates ideas from two existing frameworks, notably Soar (Newell, 1990) and ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere & Qin, 2004). Figure 1 shows the components of this generic architecture, which includes a long-term memory encoded as production rules, which matches against the current goal stack and the contents of short-term memory. The rules are strengthened when they match or when they resolve problems within the goal stack, such as achieving or filling in some goal. A structural learning mechanism adds

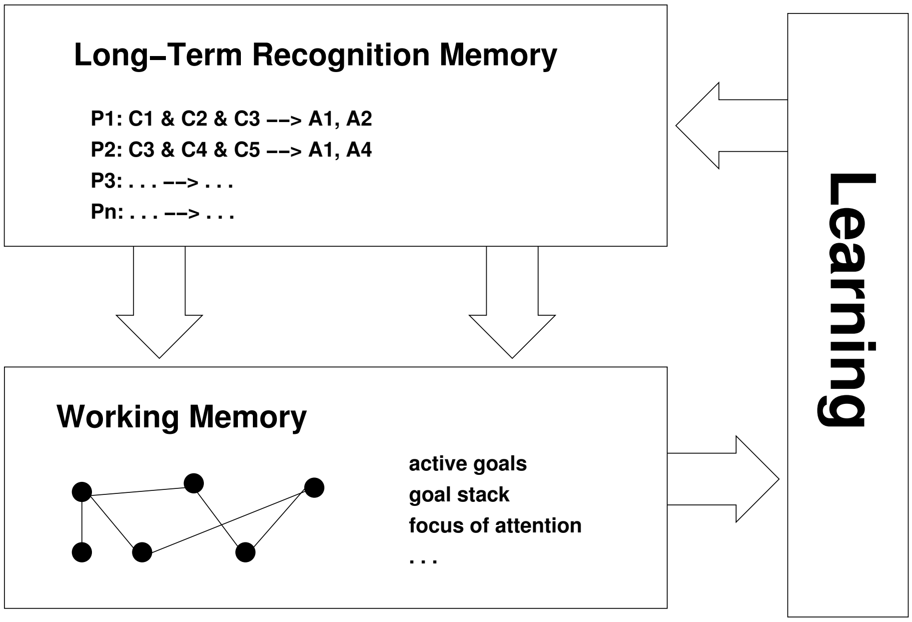

Figure 1. The components of a prototypical architecture.

new rules to long-term memory, which alters future performance.

3.1 The simple lights and buttons task

To illustrate some of these concepts, consider a task that involves pushing buttons underneath lights that are on. A simple version uses four lights and four buttons, two per hand. If a light is on, the subject presses the corresponding button, as shown in Figure 2. In total, there are 16 different patterns of lights that require 16 different responses, assuming that not pushing all four buttons is a possible response. Seibel (1963), who studied a more complex version of this task, found that subjects become faster at this task the longer they practiced it.

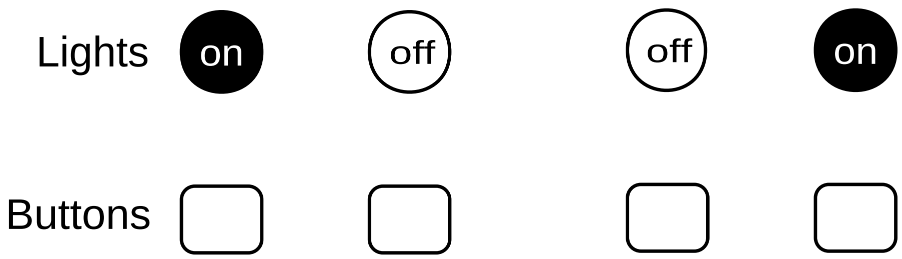

Figure 2. An example task that involves pressing buttons where the light is on.

3.2 The model

Our model in the prototypical architecture assumes a hierarchical decomposition of the task and of the knowledge to perform it. To master the task with both hands, subtask for each hand, mastery for each button must be achieved; to master a button, the light must be checked and the appropriate action done to the button (pressed if the light is lit, or not pressed if it is off). Figure 3 depicts this organisation by grouping the lights by pairs. We have based this simple model on a previous one (Newell & Rosenbloom, 1981; Rosenbloom & Newell, 1987).

The model predicts that the time taken to press the appropriate lights depends on the production rules available in long-term memory, and also that learning occurs in a cumulative manner, in that combined responses across fingers can be acquired only when the subresponses are known. The lowest level response, for an individual light, is atomic and thus always known. Learning combined responses requires acquiring new rules denoted by nodes e, f, and g in Figure 3. At the outset, the first set of lights thus takes a total of seven steps. The model takes one step to do each of the four lights, one step to compose each of the two pairs, and one

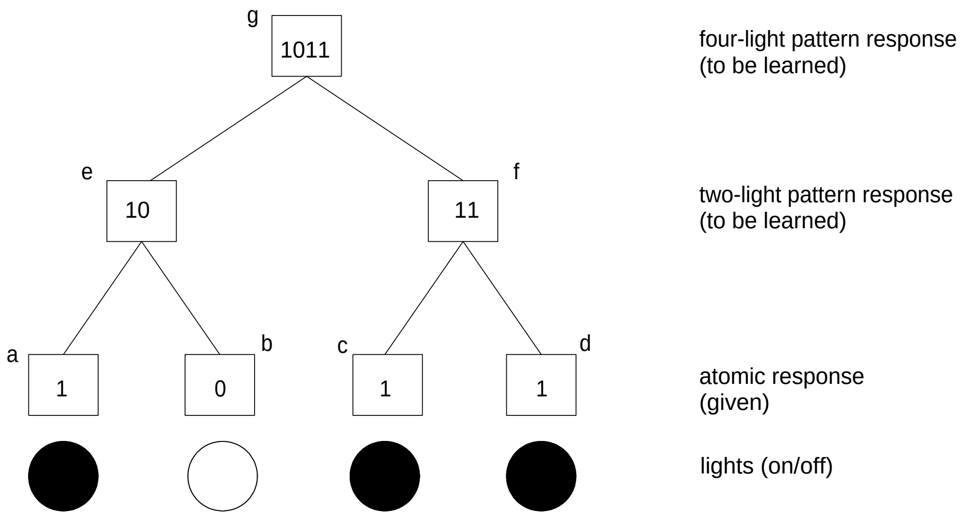

Figure 3. A simple example model of the Seibel task.

step to compose the four-light set. Recognizing each pair saves two steps, and recognizing the whole four-light set saves six steps.

Defining the task knowledge and how it is used lets us describe how to create a good learning sequence. An efficient training order for this model draws as much as possible on what is known and provides opportunities for acquiring the maximum number of rules each time. In contrast, a poor learning sequence provides less opportunities for learning. For instance, repeatedly practicing the same two-light pattern on one of the hands while the other hand learns a new two-light pattern does not lead to acquistion of higher level knowledge (four-light pattern) as quickly as with the more efficient order where new patterns are learned by both hands each trial.

Table 1 gives an example of an efficient and less efficient learning sequence for the task. In general, the most efficient approach for this architecture on this task, at least with this knowledge and representation, is to keep the learning mechanism busy. Less efficient sequences let the subtasks be repeated without learning, particularly precluding higher level learning occurring. For example, when the remaining eight unseen items are presented, the learner with the bad sequence must be shown three additional trials to catch up with the efficient sequence, giving a total of 19 instead of 16 trials, which amounts to a 16% slower learning rate.

Table 1. How two different sequences with the same items but in different order can lead to different learning.

An efficient sequence for learning			Less efficient sequence for learning		
Stim.#	Stimuli	Patterns learned	Stim.#	Stimuli	Patterns learned
0	oo oo	ooL ooR	1	oo ox	ooL oxR
5	ox ox	oxL oxR	2	oo xo	xoR
10	xo xo	xoL xoR	3	oo xx	xxR
15	xx xx	xxL xxR	0	oo oo	ooR
1	oo ox	ooox	4	ox oo	oxL
2	oo xo	ooxo	5	ox ox	oxox
3	oo xx	ooxx	10	xo xo	xoL
4	ox oo	oxoo	15	xx xx	xxL
Learned:	8 two-light patterns		Learned:	8 two-light patterns	
	4 four-light patterns			1 four-pattern	
% learned	50%		% learned	37.5%	

Note. There are 24 things to learn in this simple world (8 two-light patterns and 16 four-light patterns). The symbol 'x' indicates lights that are on, whereas 'o' indicates lights that are off. Stimuli numbers are based on the set of 16 different four-light patterns.

Despite the simplicity of both task and model, they are complex enough to exhibit order effects. Moreover, the task is similar to many others that occur in the real world in which component knowledge must be mastered before more advanced knowledge can be acquired. This effect can be found in existing Soar models of job shop scheduling (Nerb, Ritter, & Krems, 1999), blood typing (Johnson, Krems, & Amra, 1994), circuit troubleshooting (Ritter & Bibby, 2001), and language acquisition (Lewis, 1998).

This simple example shows that several things—the structure of the task, the internal representation, the performance process, the order of stimuli, and the learning mechanism—interact to create order effects. A complete account of such effects must specify each of these components.

4 Aspects of process models that can explain order effects

Ideally, process models do not just explain but also predict data, including the effects of training order. A well-crafted computational account of human behavior can suggest novel conditions under which phenomena of interest occur. One advantage of casting models within a constrained theory of the cognitive architecture is that they are more likely to produce such predictions.

Order effects can also arise from processing that leaves changes in the process model's state or mechanisms that interact with later processing like soap left on poorly rinsed pots that influence later cooking. In this section we consider five possible factors within cognitive architectures that can explain and predict such effects: (a) forgetting, (b) not forgetting, (c) memory overload, (d) changes to the internal processes, and (e) time constraints that arise in rapidly changing environments. Undoubtedly, other sources of explanation are possible, but these seem likely to account for many cases in which training order influences learning.

4.1 Forgetting

Order effects can occur whenever structures cannot be retrieved later, say, due to decay over time or interference from other structures. A model may also assume that activations decay over time or that retrieval can fail due to shifts of attention. Order effects appear when an interaction between a pair of cognitive elements that facilitates or hinders learning of these elements, when those elements can get forgotten over time, or when the time interval between processing the elements is not fixed. For example, suppose one is learning about a country A and its capitol B. Knowing A facilitates learning B and vice versa, but facilitation occurs only if A is still active in memory when B is learned. Because forgetting is a function of time, sequences in which A and B appear close to each other should produce faster learning than ones in which A and B are distant. Pavlik's chapter (Chapter 10) explores this idea in some detail.

Many learning models rely on the cooccurence of elements in dynamic memories. For example, the composition mechanism in ACT-R (Anderson, 1993; Jones et al., 2000) uses information about the elements matched by successively applied rules to create a new rule. Similarly, the chunking mechanism in Soar (Newell, 1990) uses dependencies among elements in working memory before and after it solves a problem to determine the content of new production rules. Finally, recurrent neural networks (Elman, 1989) change weights on links only to the extent that the nodes from which they emanate were active. Factors that influnce the retrieval of either of such pairs will also influence learning.

4.2 Einstellung (not forgetting)

Another source of order effects is reliance on strategies that have proved successful in the past. Once a person has found some way to solve a problem, he often continues to utilize the same solution even when other responses may work better in new situations. Such *Einstellung* behavior occurs more often when the person encounters problems in certain orders.

The classical example for such effects are sequences of arithmetical water jar puzzles. Luchins (1942) showed that, depending on the order in which subjects solved such problems, subjects used more or less efficient strategies to solve later problems (for a more recent demonstration of the effect see Luchins & Luchins, 1991). If an individual is given a series of problems (Set A) for which he acquires a strategy that works, he tends to use the same strategy on other problems (Set B) even when a simpler solution is possible. However, if a person is presented with problems from Set B first, he nearly always finds the more elegant solution.

Cognitive scientists have developed a number of process models for Einstellung that incorporate various learning mechanisms. These composing sets of production rules into larger ones (Neves & Anderson, 1981) and analogical reasoning based on earlier solutions (Gick &

Holyoak, 1980; Jones & Langley, in press). Scheiter and Gerjets (Chapter 14) examine a related concept. What these models have in common are methods that create new long-term structures from individual experiences, which they then prefer to utilize in new situations rather than carrying out search for more efficient solutions.

4.3 Cognitive overload

A third factor that can underlie effects of training order is cognitive load. Sweller (1988, 1994, Chapter 14) has developed the most extensive theory of how demands on working memory affect problem solving and learning. His account provides a framework for investigations into instructional design by considering both the structure of information and the cognitive processes that let learners interpret that information. The theory assumes that working memory demands stem from three additive sources: the material being learned (*intrinsic* cognitive load); the manner in which information is presented (*extraneous* or *ineffective* load); and resources devoted to learning and automation (*germane* or *effective* cognitive load).

In this framework, a well-designed learning sequence maximizes resources that can be devoted to germane cognitive load. Intrinsic cognitive load depends on the complexity of the material being learned, being high if relevant elements interact with each other and low if they can be learned independently. Effective instructional design therefore minimizes extraneous cognitive load during learning, as well as the load from interfaces themselves. In series of studies, Sweller and his colleagues have shown how different cognitive loads imposed by external structuring (including ordering) of tasks can facilitate or hinder learning. Of crucial interest here are the kinds and timing of instructions and learning aids that will encourage effective learning (Bass, Baxter & Ritter, 1995).

A further source for order effects within this framework has to with intrinsic cognitive load, which was initially considered as given and static as well as irreducible. A more recent view, however, assumes that intrinsic cognitive load may itself be a function of the task-subject interaction. In particular, the learner's level of expertise may correspond to alterations in intrinsic cognitive load. An interesting outcome of research in this area is the so called *expertise reversal effect* indicating that instructional techniques that are effective with novices can become ineffective when used with more experienced learners (Kalyuga, Ayres, Chandler & Sweller, 2003, Renkl & Atkinson, Chapter 7). A recent overview about both the theoretical and empircal status of the theory are given by Pass, Renkl and Sweller (2003, 2004).

4.4 Changes to internal processes

At the moment, most architectures provide the same mechanisms and information processing capabilities across situations and time. This assumption is at odds with common sense and, more importantly, with findings from psychological research. Fatigue is a simple and intuitive example of how the way we process information changes: we process information differently in a vigilant than in a tired state. Likewise, motivational and emotional states can change perception and information processing. Order effects can arise because different sequences of training material can lead to different emotional or motivational states. For instance, a sequence might be more or less boring for learners influencing their motivation for further learning and problem solving.

The work of Isen (2000) provides psychological evidence for effects of mood on information processing. She showed that different moods lead to different success in problem solving and learning. Positive mood leads to more creativity (or at least more variance in behavior) and negative mood leads to more accurate behavior. A good mood lets the problem solver work more flexibly, whereas a bad mood makes the problem solver eager to get positive reinforcement as soon as possible. The order of problems can lead to these modes. For example, difficult tasks early in a sequence can lead to frustration and other negative emotional states (see Bower, 1981; Nerb & Spada, 2001; Thagard & Nerb, 2002, for further psychological evidence that cognitions influence and are influenced by emotions).

Recently however, we are experiencing a growing interest in the study of emotions within Cognitive Science and expecially within Artificial Intelligence (e.g., Cañamero & Hudlicka, 2004; Minsky, in prep.; Norman, Ortony & Russell, 2003; Picard, 1997; Silverman, 2004). As a result of these efforts, some computational models about the role of emotions during reasoning and problem solving have emerged; example models are Cathexis (Velásquez, 1998), EMA (Gratch & Marsella, 2004), HOTCO

(Thagard, 2003), and DEBECO (Nerb, 2004). None of these models are concerned explicitly with order effects. However, order effects in learning will result if within these models emotions have lingering effects on cognitive mechanims. Altogether, this is a fascinating new development and a growing field of research that has huge potential for changing the study of learning and problem solving (Kort, Reilly & Picard, 2001).

4.5 Time constraints in rapidly changing environments

The order in which a learner attempts tasks is often important in highly interactive, externally paced environments that require timely responses. If the situation is novel enough or if the response is complicated enough, he will not be able to respond before the situation changes and he loses a chance to learn from that context. For example, in video games, the player must hit a target before it disappears. If he does do not accomplish this in time, then he may not be able to acquire even a partial response and the task can remain impossible.

One way to avoid this problem is to change the order of tasks. If the learner encounters easier situations before harder ones, then he will be able to respond more often and thus able to learn from the outcomes. Moreover, learning on easy tasks can reduce the time needed to respond in more complicated situations, thus allowing responses and learning on them as well. For this reason, such *part-task training*, which presents component tasks before the complete task, is a common approach to training in real-time environments (Donchin, 1989).

This account predicts part-task training would be beneficial in the lights and buttons domain if the button pushes had a deadline. When a task required pushing many buttons, a novice would not initially be able to respond in time. However, presenting tasks that involve single light, followed by those with pairs of lights, and so forth up to the full set would allow learning on early trials. This approach would support learning on the later trials because the partial responses would be available for combination into more complex ones.

5 From concrete models to abstract models

As we have seen, process models in cognitive science typically take the form of a running AI system that performs some task, and that also is constrained to carry our the task in much the same way as humans. However, two problems can arise with this approach that can be solved by using a somewhat different form of process model.

The first problem is the difficulty in creating a complete AI model, which is not always straightforward. Developing a system that performs the task requires that one specify a complete algorithm and a way for the model to interact with the task. The modeler must have substantial programming skills. The model must also be run, which is usually straightforward, but which can be time consuming if one applies it to a wide range of tasks or if one desires expected or average behavior. One may also have to interpret or code the results of the model's behavior.

The second problem is that, in order to develop a complete, running model, one must often make arbitrary assumptions about structures or processes that have no theoretical content. Even when the model fits the data well, its success may not be due to its core theoretical assumptions but rather to its arbitrary components. An example of this occurred with context effects in letter perception, where the encoding of the stimuli by the modellers appears to have been more important than the processes used in the model itself (McClelland & Rumelhart, 1981; Richman & Simon, 1989).

A response to these difficulties is to develop an *abstract* process model of the phenomena. Unlike the 'concrete' information processing models we have been discussing, an abstract model makes fewer commitments about structures and processes, which means that it cannot actually perform the task, but rather represents behavior at a coarser level. Whereas concrete models characterize what the human will do and the reasons, abstract models are often used to predict quantitative measures such as the time to make choices and how much is learned.

Cognitive scientists have developed abstract models of learning and problem solving (Atwood & Polson, 1976; Ohlsson, 1995; Schooler & Hertwig, 2005), sensory-motor behavior (Langley, 1996), categorization (Langley, 1999), and decision making (Young, 1998), but one of the earliest is due to Rosenbloom and Newell (1987) for the lights and buttons (Seibel) task described above. They initially presented a concrete production system model to explain the power law of practice (how reaction times decrease with practice but at a decreasing rate). For reasons of computational efficiency, they used a abstract model based on the production system model to compute the reaction times for a series of trials for comparison with the human data.

Abstract models can be much easier to use. Later work by Ritter (1988) utilized another abstract model to compute the expected value for each trial on this task. In each case, the predictions of the abstract model were easier to manipulate, and could be derived around 100,000 times faster (five seconds vs. 100 runs $\times$ 5 hours per run $\times$ 3,600 seconds/hour), than the concrete rule-based model that actually performed the task.

One drawback of abstract models as typically used is that they average over different training sequences and thus cannot account for order effects. However, there is nothing inherent in the abstract approach that forces such averaging over all tasks. If one's goal is average learning curves, they may be the only practical way to achieve this end. For example, Nerb, Ritter, and Krems (1999) report a concrete model of behavior on more complex problem-solving task that provides reasonable timing predictions for sequences of problems for individual subjects. Each trial takes over a minute to run, with learning taking place over a sequence of 25 trials, and a hundred runs would be required to achieve reliable average. An abstract modeling approach could achieve this result far more efficiently.

To illuminate this point, let us consider an abstract model that captures the essential features of the concrete lights and buttons model.

Rather than running the concrete model repeatedly to compute the expected average time across trials for the first two test trials, Trials 9 and 10, we can compute the expected time assuming a uniform distribution of all the possible stimuli.

As shown at the top of Table 2, the efficient sequence starts with a better knowledge base. It can recognize each of two-light patterns and a quarter of the four-light patterns. The inefficient sequence can recognize all the sub patterns, but not as many larger patterns. In trial 9, the efficient sequence has a greater chance of applying a four-light pattern than the inefficient sequence. The inefficient sequence, on the other hand, has a greater chance to learn a new pattern. This effect carries into trial 10. With repeated trials, the two will converge, with the more efficient sequence always being faster, but by an ever decreasing amount.

This abstract model provides several lessons. First, it illustrates how seemingly similar but theoretically different sequences can lead to different learning. Second, it illustrates how an abstract model can be used, and how easily they can be used to represent a process model and its predictions.

6 Concluding remarks

Although process models of human behavior, including learning, have existed for almost five decades, considerable work still remains. In closing, we briefly consider some open questions with respect to computational

Table 2. Expected time for stimuli 9 and 10 if they are presented randomly, along with the stored knowledge after the training sequences in Table 1.

After the efficient sequence there are in the model: 8 two-light patterns 4 four-light patterns	After the less efficient sequence there are in the model: 8 two-light patterns 1 four-light pattern
On Trial 9:	
[no learning situation] 4 / 16 four-light patterns known × 1 model cycle if matched [learning situation] 75 % chance of learning a new four-light pattern 12 / 16 unknown four-light patterns × 3 model cycles (two-light patterns)	[no learning situation] 1 / 16 four-light patterns known × 1 model cycle if matched [learning situation] 93 % chance of learning a new four-light pattern 15 / 16 unknown four-light patterns × 3 model cycles (two-light patterns)
2.5 model cycles expected response time (.25 × 1 cycle) + (.75 × 3 cycles)	2.97 model cycles expected response time (.065 × 1 cycle) + (.925 × 3 cycles)
After Trial 9: 8 two-light patterns 4.75 four-light patterns	8 two-light patterns 1.93 four-light patterns
On Trial 10:	
[no learning situation] 4.75 / 16 patterns known × 1 model cycle if all matched [learning situation] 70 % chance of learning a new four-light pattern 11.25 / 16 unknown patterns × 3 model cycles (two-light patterns)	[no learning situation] 1.93 / 16 pattern known × 1 model cycle if all matched [learning situation] 88 % chance of learning a new four-light pattern 15 / 16 unknown patterns × 3 model cycles (two-light patterns)
2.4 model cycles expected response time (.30 × 1) + (.70 × 3)	2.76 model cycles expected response time (.12 × 1) + (.80 × 3)

models of order effects that the reader may wish to explore on his own.

6.1 Experimental tests of predictions

One advantage of process models is that they let one make predictions about behavior in new situations, which can then suggest additional experiments that can either support or disconfirm the model. Some useful activities of this sort would include:

- Identify situations in which the model's predictions about the effects of training order disagree with common sense and design an experiment to determine which is correct. Results that agree with the model provide more compelling evidence than ones for studies that simply agree with intuitions. For example, in the lights and buttons task, consider what would happen if half the stimuli were not presented until after extensive practice (Simon, personal communication, 1988). Most models would consider them as fairly new stimuli, but would human subjects treat them in this way?
- Identify situations in which competing models make different predictions about order effects and design an experiment to discriminate between them. Such studies tell more about the nature of human learning than ones in which the models agree. For example, in the lights and buttons task, one could create two different problem representations. If the two reprepresentations predict different behavior an experiment might help resolve the

apparent contridiction.

- Identify aspects of a process model that explain order effects (as discussed in section 4) and design experiments that vary task characteristics to determine which aspects are responsible. Such studies can lead to gradual refinement of process models that can make increasingly specific predictions.

- Identify situations in which a process model indicates that the learner's background knowledge will mitigate or eliminate order effects and design experiments to test this prediction. Such studies can reveal more information about the role of expertise in learning than experiments focusing on simple novice to expert transitions. In the lights and buttons task, one might expect pianists to exhibit weaker order effects because they have extensive knowledge about keys.

Of course, these types of experiments are not specific to the study of order effects; they can be equally useful in understanding other aspects of human behavior. But the empirical study of sequencing has been so rare, in particular in the context of evaluating process models, that they seem especially worthwhile.

6.2 Developing models and architectures

Because there are relatively few process models of order effects, another important activity is the creation and refinement of such models. Some likely work of this sort would include:

- Identify a simple order effect and develop a model that explains it. For example, create a model that explains the benefits of part-task training (Mane & Donchin, 1989), which emphasises teaching the component skills of a task before teaching how to integrate them. After creating the model, consider what suggestions it makes for instruction in the area. The model need not be concrete, but it should be clear enough to predict implications like relative learning rates.
- Identify an order effect that has not yet been explained and develop a concrete process model that explains it within an existing architectural framework. An even better approach would involve modelling the effect within different architectures and, if they share underlying features, designing an abstract model that subsumes them.
- Identify places in an existing architecture where introduction of resource or timing limitations would suggest new order effects, then develop concrete models that instantiate this prediction for a specific task or set of tasks.

Again, these types of activities apply to any class of psychological phenomena, but order effects have received so little attention that they seem an especially fertile area to use in constructing and constraining our theories of the human cognitive architecture.

6.3 General advice

In addition to the above suggestions on open problems in the study of order effects, we can offer some general advice to erstwhile cognitive modellers. First, we encourage researchers to select a theoretical framework, ideally one that takes a clear position on the nature of the human cognitive architecture, and to develop models within that framework. If researchers are new to the area, then they should not work in isolation, but rather attach themselves to a scientist or group experienced with using that framework. At the same time, they should not focus their attention on this framework to the exclusion of all others; understanding alternative theories and their relation to one's own is also part of the scientific process.

Second, computational modellers should also remember that it is essential to relate their systems to phenomena. A model should always make contact with observations of some sort. Moreover, like other disciplines, cognitive science operates not by attempting to confirm its theories, but rather by gaining insights about ways to improve them (Grant, 1962). The construction, evaluation, and analysis of process models is an important means to this end.

Acknowledgements

Support for this work has also been provided to the second author by the DERA, contract number 2024/004, by the ESRC Centre for Research in

Development, Instruction and Training, and ONR contracts N00014-03-1-0248 and N00014-02-1-0021. The views expressed here are those of the authors and an acknowledgement here does not indicate an endorsement of this report from those sponsors listed.

References

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C. & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*, 1036–1060.

Anderson, J. R. & Gluck, K. (2001). What role do cognitive architectures play in intelligent tutoring systems? In D. Klahr & S. M. Carver (Eds.), *Cognition & Instruction: Twenty-five years of progress* (pp. 227–262). Mahwah, NJ: Erlbaum.

Atwood, M. E. & Polson, P. G. (1976). A process model for water jug problems. *Cognitive Psychology*, *8*, 191–216.

Bass, E. J., Baxter, G. D. & Ritter, F. E. (1995). Using cognitive models to control simulations of complex systems. *AISB Quarterly*, *93*, 18–25.

Bechtel, W. & Abrahamsen, A. (2002). *Connectionism and the mind: Parallel processing, dynamics, and evolution in networks, (2nd ed.)*. Malden, MA: Blackwell Publishers.

Bower, G. H. (1981). Mood and memory. *American Psychologist*, *36*, 129–148.

Cañamero, L. & Hudlicka, E. (2004). *Architectures for Modeling Emotion: Cross-Disciplinary Foundations. AAAI Spring Symposion Series*. Menlo Park, CA: AAAI.

Daily, L. Z., Lovett, M. C. & Reder, L. M. (2001). Modeling individual differences in working memory performance: A source activation account in ACT-R. *Cognitive Science*, *25*, 315–353.

Donchin, E. (1989). The learning strategies project: Introductory remarks. special issue: The learning strategies program: An examination of the strategies in skill acquisition. *Acta Psychologica*, *71(1-3)*, 1–15.

Elman, J. L. (1989). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, *7*, 195–225.

Feigenbaum, J. R. & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, *8*, 305–336.

Gick, M. L. & Holyoak, K. J. (1980). Analogical problem solving. *Cogntive Psychology*, *12*, 306–355.

Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models. *Psychological Review*, *69*, 54–61.

Gratch, J. & Marsella, S. (2004). A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, *5*, 269–306.

Isen, A. (2000). Positive affect and decision-making. In M. Lewis & J. M.

Haviland (Eds.), *Handbook of Emotions (2nd ed.)* (pp. 417–435). New York: Guilford.

Johnson, T. R., Krems, J. & Amra, N. K. (1994). A computational model of human abductive skill and its acquisition. In *Proceedings of the Annual Conference of the Cognitive Science Society* (pp. 463–468). Mahwah, NJ: Erlbaum.

Jones, G., Ritter, F. E. & Wood, D. J. (2000). Using a cognitive architecture to examine what develops. *Psychological Science*, *11*, 93–100.

Jones, R. M. & Langley, P. (in press). A constrained architecture for learning and problem solving. *Computational Intelligence* (pp. xx–xx).

Kalyuga, S., Ayres, P., Chandler, P. & Sweller, J. (2003). The expertise reversal effect. *Educational Psychologist*, *38*, 23–32.

Kort, B., Reilly, R. & Picard, R. W. (2001). An affective model of interplay between emotions and learning: Reengineering educational pedagogy—building a learning companion. In Okamoto, T., Hartley, R. & Kinshuk, J. P. K. (Eds.), *International Conference on Advanced Learning Technology: Issues, Achievements and Challenges* (pp. 43–48). IEEE Computer Society.

Langley, P. (1996). An abstract computational model of learning selective sensing skills. In *Proceedings of Eighteenth Annual Conference of the Cognitive Science Society* (pp. 385–390). Hillsdale, NJ: Erlbaum.

Langley, P. (1999). Concrete and abstract models of category learning. In

Proceedings of the Twenty-First Annual Conference of the Cognitive Science Society (pp. 288–293). Mahwah, NJ: Lawrence Erlbaum.

Lewis, R. L. (1998). Leaping off the garden path: Reanalysis and limited repair parsing. In J. D. Fodor & F. Ferreira (Eds.), *Reanalysis in sentence processing* (pp. 247–284). Boston, MA: Kluwer Academic Publishers.

Luchins, A. (1942). Mechanization in problem solving. *Psychological Monographs*, *54*.

Luchins, A. S. & Luchins, E. H. (1991). Task complexity and order effects in computer presentation of water-jar problems. *The Journal of General Psychology*, *118*, 45–72.

Mane, A. M. & Donchin, E. (1989). The space fortress game. special issue: The learning strategies program: An examination of the strategies in skill acquisition. *Acta Psychologica*, *71(1-3)*, 17–22.

McClelland, J. L. & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*, 375–407.

Minsky, M. (in prep.). *The Emotion Machine*. Retrieved November 6, 2005, from http://web.media.mit.edu/∼minsky/.

Neches, R., Langley, P. & Klahr, D. (1987). Learning, development, and production systems. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learning and development* (pp. 1–53). Cambridge, MA: MIT Press.

Nerb, J. (2004). How desire coherence and belief coherence lead to emotions: A constraint satisfaction model of the appraisal process. In Cañamero, L. & Hudlicka, E. (Eds.), *Architectures for Modeling Emotion: Cross-Disciplinary Foundations. AAAI Spring Symposion Series* (pp. 96–103). Menlo Park, CA: AAAI.

Nerb, J., Ritter, F. E. & Krems, J. (1999). Knowledge level learning and the power law: A Soar model of skill acquisition. *Kognitionswissenschaft [Journal of the German Cognitive Science Society]*, *8*, 20–29. Special issue on cognitive modelling and cognitive architectures, D. Wallach & H. A. Simon (eds.).

Nerb, J. & Spada, H. (2001). Evaluation of environmental problems: A coherence model of cognition and emotion. *Cognition and Emotion*, *15*, 521–551.

Neves, D. M. & Anderson, J. R. (1981). Knowledge compilation: Mechanisms for the automatization of cognitive skills. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 57–84). Hillsdale, NJ: Erlbaum.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, Ma: Harvard University Press.

Newell, A. & Rosenbloom, P. S. (1981). Mechanism of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 1–56). Hillsdale, NJ: Erlbaum.

Newell, A., Shaw, J. C. & Simon, H. A. (1962). The process of creative thinking. In H. E. Gruber, G. Terrell & M. Wertheimer (Eds.), *Contemporary approaches to creative thinking* (pp. 63–119). New York: Lieber–Atherton. Reprinted in H.A. Simon's 1979 *Models of Thought*, New Haven, CT: Yale Univeristy Press.

Norman, D. A., Ortony, A. & Russell, D. M. (2003). Affect and machine design: Lessons for the development of autonomous machines. *IBM Systems Journal*, *42*, 38–44.

Ohlsson, S. (1995). Abstract computer models: Towards a new method for theorizing about adaptive agents. In N. Lavrac & S. Wrobel (Eds.), *Machine Learning: ECML-95* (pp. 33–52). Berlin: Springer.

Paas, F., Renkl, A. & Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. *Educational Psychologist*, *38*, 1–4.

Pass, F., Renkl, A. & Sweller, J. (2004). Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture. *Instructional Science*, *32*, 1–8.

Picard, R. W. (1997). *Affective Computing*. Cambridge, MA: MIT Press.

Richman, H. B. & Simon, H. A. (1989). Context effects in letter perception: Comparison of two models. *Psychological Review*, *96*, 417–432.

Ritter, F. E. (1988). Extending the seibel-soar model. In *Proceedings of the Soar V Workshop Carnegie Mellon University*.

Ritter, F. E. & Bibby, P. (2001). Modeling how and when learning happens in a simple fault-finding task. In *Fourth International Conference on Cognitive Modeling* (pp. 187–192). Mahwah, NJ: Erlbaum.

Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R. M., Gobet, F. & Baxter, G. D. (2003). *Techniques for modeling human performance in synthetic environments: A supplementary review.* Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center (HSIAC).

Rosenbloom, P. S. & Newell, A. (1987). Learning by chunking: A production system model of practice. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learning and development* (pp. 221–288). Cambridge, MA: MIT Press.

Schooler, L. J. & Hertwig, R. (2005). How forgetting aids heuristic inference. *Psychological Review*, *112*, 610–628.

Seibel, R. (1963). Discrimination reaction time for a 1,023-alternative task. *Journal of Experimental Psychology*, *66*, 215–226.

Silverman, B. G. (2004). Human performance simulation. In J. W. Ness, D. R. Ritzer & V. Tepe (Eds.), *The science and simulation of human performance* (pp. 469–498). Amsterdam: Elsevier.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*, 257–285.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning & Instruction*, *4*, 295–312.

Thagard, P. (2003). Why wasn't O. J. convicted? emotional coherence in legal inference. *Cognition and Emotion*, *17*, 361–383.

Thagard, P. & Nerb, J. (2002). Emotional gestalts: Appraisal, change, and emotional coherence. *Personality and Social Psychology Review*, *6*, 274–282.

Velásquez, J. D. (1998). Modeling emotion-based decision-making. In *Proceedings of the 1998 AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition*. Orlando, FL: AAAI Press.

Young, R. M. (1979). Production systems for modeling human cognition. In D. Michie (Ed.), *Expert systems in the micro-electronic age* (pp. 35–45). Edingburgh, UK: The University Press. Reprinted in E. Scanlon & T. O'Shea (Eds). *Educational computing*, Wiley, 1987. 209-220.

Young, R. M. (1998). Rational analysis of exploratory choice. In M. Oaksford & N. Chater (Eds.), *Rational models of cognition* (pp. 469–500). Oxford (UK): Oxford University Press.