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

Frank E. Ritter, Pat Langley, Josef Nerb

February 17, 1998

Frank says: ________________________________

Notes and to do list:

• 2000-5000 word limit, for the LHM task-force 3 book.

Contents

1 Introduction 3

2 Process models in cognitive science 4
   2.1 The advantages of formal models ......................... 4
   2.2 Types of process models ................................. 5

3 A simple architecture and model showing an order effect 7
   3.1 A prototypical architecture .............................. 7
   3.2 The simple lights and buttons task ..................... 8
   3.3 The model ............................................. 8

4 Aspects of process models that explain order effects 10
   4.1 Forgetting ............................................. 10
   4.2 Not forgetting ......................................... 11
   4.3 Cognitive overload .................................... 11
   4.4 External response-time deadlines ....................... 12
   4.5 Changes to internal processes ........................ 12

File chapter from February 17, 1998
5 From concrete models to abstract models

6 Conclusions and open questions
   6.1 Experimental tests of predictions 
   6.2 Developing models and architectures 
   6.3 General advice
This chapter is part of a book designed to increase awareness of and interest in order effects in learning. It is designed for late undergraduates, interested lay people such as educators, and to serve as a source of inspiration for graduate students such as Mindstorms.

**Keywords:** Process models, abstract models.

**Frank Ritter** +44 115 951-5292, -5234 (FAX) Frank.Ritter@nottingham.ac.uk

**Josef Nerb** +49 761 203-2497, -2496 (FAX) nerb@psychologie.uni-freiburg.de

**Pat Langley** +1 415 2532/2500, -2555 (FAX) langley@rtma.daimlerbenz.com

**Abstract**

To fully understand sequential effects on learning in humans, we will have to have a rather complete theory of cognition. The theory will be complete enough that it must perform the task of interest and learn like humans while doing so. These theories, called process models, can be broken down into two parts, the aspects of the model that do not change between tasks — the architecture — and the aspects that do change across tasks — the knowledge. Where these models have been used to understand sequential behaviour and sequence effects on learning, they can be very powerful. We present, as an example, a simple model of a simple task that shows how an optimal order can lead to significantly (16%) faster learning. Process models, although powerful, remain difficult to apply routinely. In response we discuss an alternative approach — developing 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 not static but involves change over time, more complete explanations of cognition often take the form of process 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, including those from machine learning, and computational models of human behavior, including some illustrations of the psychologically more plausible models. After this, we present a computational model of order
effects on a simple task, cast within a particular (but simplified) theoretical framework. Next, we explore more broadly 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 a 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, that is, the how and what of the process that makes up the system. For example, physics often uses the formalism of differential equations to describe the relationship between quantitative variables (say, heat and temperature in a furnace) at one moment in time and the next. Process models of human behavior have somewhat different requirements as what changes over time are not (solely at least) continuous variables, but rather qualitative structures in short-term and long-term memory (knowledge or information).

Fortunately, computer languages provide a formalism that can be used for modeling human behavior in the same way that differential equations are used in physics. They do this by describing how symbolic structures change over time in information processing terms. Moreover, the field of AI has developed a variety of representations, performance elements, and learning methods that can operate on many of the tasks that confront humans. Some AI work has little to say for cognitive science because it makes no effort to constrain its method to match psychological phenomena, but other AI systems have been developed with this 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 process that generate behavior lead to specific predictions and thus can be shown to be incorrect (models that can perform a task can also be put to work, opening many possibilities for applications). Such results can thus lead to improved models that account for problematic findings and that also make new predictions,
leading to iterative progress towards more and more complete theories. They also let us examine the internal states and mechanisms in the model that gave rise to the observed behaviour, such as order effects. This lets us predict the effects of different conditions, such as different orders, without running subjects.

Another advantage of having a model's behaviour to hand is that it provides a reference for analysing a subject's behaviour — it lets us partition behavior into portions that the model can explain and those that it cannot, thus identifying anomalous behavior. This in turn will help indicate where the model is incorrect and suggest where to improve it. Finally, a well developed, parameterized model can be used to classify subjects by their characteristics (e.g., Lovett, Reder & Lebiere, 1997).

2.2 Types of process models

Before examining how computation models can explain order effects observed in human learning, we must briefly review the major types of models. Early process models had their behavior compared to data, but their internal structures were not often shared and often rather idiosyncratic. In the last decade, researchers have imposed an increasing number of theoretical constraints across sets of models. These often take the form of cognitive architectures, which characterize the structures and mechanisms of the cognitive system that are common across tasks, just as a building's architecture characterizes the general relationship between its walls, doors, roof, and patterns of use (but not the exact layout of each building). A cognitive architecture provides a language for describing the sources of behavior, including order effects.

Although there exist computational models for processes 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 (Nedches, Langley & Klahr, 1987; Young, 1979). These architectures include a long-term memory stated as condition-action rules, which changes only slowly with learning or forgetting, and a dynamic short-term (or working) memory. What varies across task models are 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, applying 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 smaller rules required extensive problem solving, the new rule can also constitute new knowledge about how to constrain future behavior.

A second well-studied framework for modeling sequential behavior is recurrent neural networks. 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.

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 determining 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 output and desired values as supervised learning) downward through the network and modifying the weights to reduce these errors.

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. There are numerous other architectures, including case-based and probabilistic ones, some of which learn and most of which do not (SIGArt, 1991; VanLehn & Ball, 1991). The architectures also differ in their assumptions about representation, knowledge retrieval, and learning, but operate in much the same way. 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 also tend to give similar effects, such as mastering simpler structures before more complex ones, despite their differences in operation. We will return to this observation later in the chapter.
3  A simple architecture and model showing an order effect

In order to consider how architectures can be used to understand order effects, and to ground the rest of the discussion in more concrete terms, we include here a worked example of a model that produces an order effect in learning. The first step is to provide a description of the model’s structure and mechanisms; then we will explain its behavior on a simple task that shows how order effects can arise.

3.1  A prototypical architecture

We assume a simple, general architecture that incorporates ideas from several existing architectures, notably Soar (Newell, 1990) and Act-R (Anderson, 1993). Figure 1 shows the components of this architecture. There is a long-term recognition memory represented as production rules, which can match against the current goal stack and the contents of short-term memory. These rules can be strengthened when they match or when they resolve problems within the goal stack, such as achieving some goal. A learning
mechanism adds new rules to long-term memory, which alters future performance.

3.2 The simple lights and buttons task

Consider a fairly simple task, which involves pushing buttons underneath lights that are on. A relatively simple version uses four lights and four buttons, two per hand. If a light is on, the subject should press the corresponding button. This is shown schematically in Figure 2. In total, there are 16 different patterns of lights requiring 16 different responses, assuming that not pushing all four buttons is a possible response as well). This is a simplified version of the Seibel task (Seibel, 1963). Over time, a long time, actually, subjects continue to get faster at this task.

![Figure 2: An example task that involves pressing buttons where the light is on.](image)

3.3 The model

Our model, based on the prototypical architecture, assumes a hierarchical decomposition of the task and the knowledge to perform it. In order to do both hands, each hand must be done and combined. In order to do each hand, each button must be done. In order to do 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). This organisation is shown in Figure 1 by grouping the lights by pairs. Other models within the architecture may be possible, and other constraints (such as how people actually learn) may be required to decide among them. We have based this simple model on a previous model (Newell & Rosenbloom, 1981; Rosenbloom & Newell, 1987).

The combined responses (across fingers) can be learned when the sub-responses are known. The lowest level response (for an individual light) is atomic and thus always known.
The performance measure considered here is the time taken to press the appropriate lights. 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 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 builds as much as possible on what is known and creates the maximum number of rules each time. A poor learning sequence has minimal learning due to repeated practice on a single pattern on one of the hands while the other hand learns a new pattern. This does not lead to higher level knowledge as quickly as the more efficient order. The sequences in Table 1 show the details, but in general the most efficient approach for this architecture on this task with this knowledge and representation is to keep the learning mechanism busy; less efficient sequences let the subtasks be repeated without learning, particularly higher level learning occurring.

Table 1: How two different sequences with the same items but in different order can lead to different learning. 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 4-light patterns.

<table>
<thead>
<tr>
<th>An efficient sequence for learning</th>
<th>Less efficient sequence for learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stim. #</td>
<td>Stimuli</td>
</tr>
<tr>
<td>0  oo  oo</td>
<td>ooL ooR</td>
</tr>
<tr>
<td>5  ox  ox</td>
<td>oxL oxR</td>
</tr>
<tr>
<td>10  xo  xo</td>
<td>xoL xoR</td>
</tr>
<tr>
<td>15  xx  xx</td>
<td>xxL xxR</td>
</tr>
<tr>
<td>1  oo  ox</td>
<td>ooxo</td>
</tr>
<tr>
<td>2  oo  xo</td>
<td>ooxo</td>
</tr>
<tr>
<td>3  oo  xx</td>
<td>ooxo</td>
</tr>
<tr>
<td>4  ox  oo</td>
<td>ooxo</td>
</tr>
</tbody>
</table>

Learned: 8 × 2 light patterns
Learned: 4 × 4 light patterns

If the remaining unseen items are presented, both sequences will pick up four-light patterns for each of them. The learner with the bad sequence must be presented with an additional three trials to catch up with the efficient sequence, for a total of 19 trials, a 16% slower learning rate.

Although simple, this task and model is complex enough to exhibit order
effects. With different semantics, this task looks like many real world tasks in which component knowledge must be learned before more complex or complete knowledge can be acquired.

This simple example shows that several things — the structure of the task, the internal task representation, the stimuli order, and the learning mechanism — interact to create order effects. In order to model such effects, one must specify each of them.

4 Aspects of process models that 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.

Generally, order effects arise from processing that leaves changes in the process model’s state that interact with later processing, like soap left on poorly rinsed pots that influence later cooking. In this section we consider five effects within a cognitive architecture order that can explain and predict such effects: (a) forgetting, (b) not forgetting, (c) no storage in memory at all, (d) changes to the basic mechanisms, 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 affects learning.

4.1 Forgetting

Order effects can occur whenever objects after their processing remain in memory and decay over time or when items stored in long-term memory cannot be retrieved later on. Many models of learning rely on the co-activation of elements in working memory or other dynamic stores. For example, the composition mechanism in Act-R (Anderson, 1993) uses information about the elements matched by successively applied rules to determine the rule it constructs. The chunking mechanism in Soar (Newell, 1990) uses the dependencies among the elements in working memory before and after it solves a problem to determine the form of new rules. And recurrent neural networks (Elman, 1989) change weights on links only to the extent that the nodes
from which they emanate were active. These architecture assume also a
decay of activations over time or problems retrieving elements due to shifts
of attention. Order effects show up when there is an interaction between a
pair of elements that facilitates or hinders learning of one or both of these
elements, when those elements can get forgotten over time, or when the time
interval between processing of the critical pair of elements is not fixed.

For example, such a pair of interacting element might be a country A
and its capitol B. Knowing A facilitates learning B and vice versa. The
facilitation effect only works if A is still active in memory when it comes to
learning B. Since forgetting is a function of time, sequences in which A and
B appear close to each other should lead to faster learning than sequences
in which A and B are distant.

4.2 Not forgetting

When subjects have found a particular strategy to work on a task, will often
continue to apply this strategy for several trials where it is not appropriate;
they lose flexibility. This kind of order effects revolve around the notion
of transfer across tasks, which Chapter TRANSFER discusses at length.
Specifically, learners may learn strategies that are transfer incorrectly to
new situations.

Luchins (1942) showed that, depending on the order in which subjects
solved water jug (pouring) problems, subjects would use more or less ef-
ficient strategies to solve later problems. This effect in problem solving—
Einstellung—is usually attributed to changes in long-term memory. Cog-
nitive scientists have developed a number of process models for this phe-
nomena that incorporate various learning mechanisms, including composing
rules into larger rules (Neves & Anderson, 1981), and analogical reasoning
based on earlier solutions (Jones, 1996; Gick & Holyoak, 1980).

4.3 Cognitive overload

The most developed theory around cognitive load effects on problem solving
and learning is Sweller’s cognitive load theory (e.g., Sweller, 1988, 1994),
CLT. The core of this theory is around how a optimum cognitive load for
problem solving and concurrent learning (schema and strategy acquisition,
automation) can be achieved. In series of studies Sweller and his colleague
have shown how different cognitive load imposed by the external structuring
(including ordering) of a task can facilitate or hinder learning and problem
solving. For a discussion of his theory for a cognitive architecture see Sweller
(1993). Of crucial interest are here in particular what kind of instructions and learning aids should be given at what time to allow for an optimum route in learning and problem solving and reflecting (e.g., Bass, Baxter & Ritter, 1995).

4.4 External response-time deadlines

The order that tasks are attempted is often important in highly interactive, externally posed environments. Timeliness of the response is necessary by definition. If the situation is novel enough, or the response is complicated enough, the learner will not be able to respond before the situation changes and they lose a chance to learn the response to that situation. For example, in a video game, when a new target appears you must hit it before it disappears (or makes you disappear). If you do not do this in time, and do not have time to learn even a partial response, (or another way to learn, such as post-task reflection) the task can remain impossible.

One way to avoid this problem is to change the order of tasks. If the tasks are presented in order from easy to hard, the learner is more likely to be able to respond (and thus to learn). The learning in the easy tasks can reduce the time to find responses to more complicated situations, allowing responses and learning in them as well. Part-task training (Donchin, 1989), which does this, presents component tasks before the complete task, is a common approach to training in real-time environments.

This effect would occur in the lights and buttons task if the button pushes had a deadline. When many buttons had to be pushed, the response would not initially be in time. Starting with single light tasks (and then pairs of lights) would allow learning on the early trials, which would support learning on the later trials because the partial responses would be available.

4.5 Changes to internal processes

As we have noted, and as assumed in the definition of architecture, most problem solving does not alter the architecture's basic mechanisms. Describing the effects (in terms of information processing) of a frustrating or emotionally charged problem might best be described as changes to the architecture (unless you prefer the architecture to include switches and settings for these effects). Fatigue is perhaps the simplest example of such effects; a difficult problem that uses up neural resources will influence later processing. Motivational and emotional states, moods, and drug-altered states can all change the preference system or the way knowledge is applied.
Different sequences of training material can lead to different emotional or motivational states. Thus, a sequence might be more or less boring for a learner, which can influence his motivation for further learning by changing the architecture’s decision procedure.

The work of Isen (e.g., Isen, Daubman & Nowicki, 1987) provides evidence that these effects can influence behaviour. 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 behaviour) 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 moods. For example, difficult tasks early in a sequence can lead to frustration.

Although we can describe these effects within the notion of a cognitive architecture, there exists very little work on modeling them. Most theories of the human cognitive architecture assume that their basic mechanisms do not change across tasks or because of tasks. But though most existing process models ignore the influence of motivation and emotions, there is no reason to believe such phenomena are beyond their scope. Wright, Sloman and Beaudion (1996) report one attempt to extend computational models in this direction (for further literature see the website, http://emotion.ccs.brandeis.edu/emotion.html), and Chapter EMOTION discusses this in greater detail.

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 actually does some task, and that also is constrained to do it in much the same way as humans do. However, problems can arise with this approach that can be solved by using an abstract version of a process model.

Creating process models with a cognitive architecture can be difficult, particularly when working with order effects. The model has to be created, which is not always straightforward. The model has to be run, which is usually straightforward, but which can be tedious if applied to a wide range of tasks. And the results of the model’s behaviour have to be interpreted.

An approach that has been used several times before receiving a formal name, is the use of abstracted, or meta-models (Ohlsson, 1995). Abstract, or perhaps better, abstracted, models are representations of a process model, but they themselves do not process information. Where process models can
tell you what will be done and why, abstracted models report measures such as how long choices would take and how much (not what) would be learned.

Abstracted models have been used to model the Seibel task several times. Rosenbloom and Newell (Rosenbloom, 1983; Rosenbloom & Newell, 1987) did it to compute responses on a trial-by-trial basis more quickly than their process model could. Later work, reported at a workshop by the first author, computed the aggregate predictions for each trial. In each case, the predictions of the abstract model were more easy to manipulate, and could be derived orders of magnitude faster (on the order of 100,000 faster, 5s vs. 100 runs x 5 hours x 3600 s/hour). Cognitive scientists have also developed abstract models of learning for problem-solving tasks (Ohlsson, 1995; Atwood & Polson, 1976) and sensory-motor tasks (Langley, 1996).

We have worked with a more complex model of a more complex problem solving task that already provides a reasonable prediction for the time of sequences of problems for individual subjects (Nerb, Krems & Ritter, 1993). Computing an averaged learning curve to see if it complies with more general psychological regularities (the power law of learning) is a natural next step for testing this model. This would not be possible by running the model repeatedly — each trial takes over a minute to run with learning taking place over a sequence of 25 trials. There would have to be several hundred runs before the data could be averaged. The only way practical way to compute an average learning curve to compare to the subject’s average, is through an abstract model.

Abstracted (based on a process model) and abstract (no process model) models are a useful methods for predicting and understanding order effects. We believe that abstracted models have a firmer position, but they both emphasise the importance of understanding behaviour over getting the bloody things to run. So we think all three are nice.

To illustrate this ability, let us consider an abstract model that captures the essential features of the concrete lights and button model. Rather than running the model repeatedly to compute the expected average time across trials for Trials 9 and 10, we can compute the expected time assuming a uniform distribution of all the possible stimuli. This analysis, in Table 2, shows that if stimuli in trials 9 and 10 are randomly selected (with replacement), the response time after the good training sequence remains faster than after the poor training sequence. With additional practice the poor sequence gradually catches up, but the expected value will never be as fast.

As shown at the top of Table 2, the efficient sequence starts with a better knowledge base. It can recognize each of 2 light patterns and a quarter of the 4 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 4-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, but the more efficient sequence will always be faster, but by
an ever decreasing amount.

There are in the learner the following pieces of knowledge after the training
sequences in Table 2.

<table>
<thead>
<tr>
<th>After the efficient sequence</th>
<th>After the less efficient sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 x 2 light patterns</td>
<td>8 x 2 light patterns</td>
</tr>
<tr>
<td>4 x 4 light patterns</td>
<td>1 x 4 light patterns</td>
</tr>
</tbody>
</table>

On Trial 9:

- [no learning situation]
  - 4 4-light patterns known/6 total
  - x 1 model cycle if all matched
- [Learning situation]
  - 2 unknown patterns/6 total
  - x 3 model cycles (2 light patterns)
  - [76 % chance of learning a
    new 4-light pattern]
  - (.25 x 1 cycle) + (.75 x 3 cycles)
  - 2.5 model cycles expected

After Trial 9.

<table>
<thead>
<tr>
<th>8 x 2 light patterns</th>
<th>8 x 2 light patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.75 x 4 light patterns</td>
<td>1.50 x 4 light patterns</td>
</tr>
</tbody>
</table>

On Trial 10:

- [no learning]
  - 4.75 patterns known/6 total
  - x 1 model cycle if all matched
- [Learning]
  - 1.25 unknown patterns/6 total
  - x 3 model cycles (2 light patterns)
  - [70 % chance of learning a
    new 4-light pattern]
  - (.30 x 1) + (.70 x 3)
  - 2.4 model cycles

| 8.12 x 1 cycle + (.92 x 3 cycles) | 2.76 model cycles |

Table 2: Expected time for stimuli 9 and 10 if they are randomly presented.

6 Conclusions and open questions

Although process models of human behavior, including learning, have ex-
isted for almost four decades, plenty of work still remains. In closing, we
briefly consider some open questions with respect to computational models
of order effects that you may wish to explore on your own. These problems range in size from an undergraduate project to PhD thesis or more. But make of them what you wish. They are useful starting points.

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 new experiments that can either support or disconfirm the model.

- Identify situations in which the predictions of a process model about training order disagree with common sense and design an experiment to determine which is correct. Such studies provide more compelling evidence for a model (if its predictions are correct) than experimental results that simply agree with intuitions. For example, in the lights and buttons task, what would happen if half the stimuli were not presented until after extensive practice (Simon, personal communication, 1988)? Most models would tend to treat them as fairly new stimuli. Would subjects treat them as new stimuli, somewhat new stimuli, or the same as the presented stimuli?

- Identify situations in which competing models make different predictions about the effects of training order 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, you could create two different representation hierarchies. If they predict different behaviour an experiment might help resolve this.

- Identify aspects of a process model that can explain order effects (as discussed in section 4 of this chapter) and design experiments that vary task characteristics to determine which aspects actually 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, we might expect pianists, who have a lot of background
knowledge, to be more immune to order effects because of their extensive knowledge about keys (or would their knowledge of chords actually hinder them?).

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, especially in the context of testing process models, that they are 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 you have created that model, consider what suggestions it makes for instruction and learning in its area. The model need not be concrete, but it should be clear enough to derive learning rates and implications.

- Identify an order effect that has not yet been explained and develop a concrete process model that explains it, preferably within an existing architectural framework. An even better approach would involve modelling the effect within two or more different architectures and, if they share underlying features, designing an abstract model that subsumes them.

- Identify places in an existing architecture where the introduction of resource or timing limitations would suggest new order effects. Then develop concrete models that instantiate this prediction for a specific task or, multiple versions of such tasks.

- Abstract models offer a possible solution, as a useful calculation aid. Could you derive a method for deriving an abstract model from process models (or a class of process models) for a given architecture? Could you go the opposite way, from an abstract or higher level model down to a more concrete model?
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 suggestions above on open problems in the study of order effects, we can offer some general advice to the erstwhile cognitive modeller. 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 the researcher is new to the area, they should not work in isolation, but rather attach themselves to a scientist or group experienced using that framework. At the same time, they should not focus their attention on this framework to the complete 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 observed phenomena; a model should model some data. Moreover, the goal of cognitive science (like other sciences) is not to confirm theories (which is best done with ambiguous and noisy data), but to gain insights on ways to improve them (Grant, 1962). This path, which requires unambiguous and clean data, is the one that leads to true progress in understanding the nature of human behavior.

Acknowledgements

Support for this work has also been provided to the first author by the DERA, contract number 2024/004, and by the ESRC Centre for Research in Development, Instruction and Training. The views expressed here are those of the authors and an acknowledgement here does not indicate an endorsement of this report from those listed.

References


