CHAPTER 10

Information Processing Approaches to Development

YUKO MUNAKATA

WHY MODEL? 427
HISTORICAL CONTEXT 429
OVERVIEW AND EXAMPLE OF EACH KIND 430
Production Systems 430
Neural Networks 432
Dynamic Systems 435
Ad Hoc Models 436
General Comparisons across Frameworks 437
DETAILED COMPARISONS: DIFFERENT MODELS OF THE SAME PHENOMENA 438

Problem Solving: The Balance Scale Task 438
Language: The Past Tense 444
Memory: The A-not-B Task 448
GENERAL ISSUES 454
Why Model? (Revisited) 454
Future Directions 456
CONCLUSION 458
REFERENCES 458

Why do children think the way they do? What leads to the remarkable changes in thinking that they show with development? What accounts for the variation in thinking observed across children? These questions are challenging, but not intractable. Developmental researchers have made great progress in addressing such questions using a vast array of methods. These include behavioral, neuroimaging, and genetic studies, with both typically and atypically developing populations.

This chapter describes a set of information processing approaches that have also supported significant contributions to the study of fundamental development questions. Broadly speaking, these approaches treat thinking as the processing of information. When applied to development, these approaches thus focus on what information children represent, how they represent and process this information, how these representations guide their behaviors, and what mechanisms lead to changes in these processes across development. Developmental change is often explained within information processing approaches in terms of self-modification processes, in which children's thinking and behaviors shape the way that they subsequently process information.

Information processing approaches have been associated with the use of formalisms that allow researchers to specify or simulate in detail the processes contributing to thinking and behavior. These formalisms may be viewed as varying along a soft-core to hard-core continuum (Klahr, 1989, 1992) Soft-core versions might include flowcharts and diagrams to describe models of children's thinking (e.g., Aguiar & Baillargeon, 2000; Case, 1986; Siegler, 1976), whereas hard-core versions are actually implemented in computational models. Computational models take the form of codes (e.g., mathematical equations and commands) that specify how a system transforms and responds to inputs; these codes can be run on a computer to observe the processing and behavior of the system under many different circumstances. As in earlier reviews (Klahr & MacWhinney, 1998), this chapter focuses on hard-core
information processing approaches. For a review of other variants of information processing approaches, see Siegler and Alibali (2005).

This chapter first addresses the question: Why should anyone care about computational models of development? The discussion explains how such models have served as a complement to other approaches of studying development, to support many advances in understanding how change occurs. The chapter then describes some of the historical context that set the stage for computational modeling endeavors. An overview and example are provided for each of four types of information processing approaches to development: production systems, neural networks, dynamic systems, and ad hoc models. Some aspects of children's development (problem solving, language, and memory) have been investigated through more than one type of information processing model, allowing direct comparisons to be made about how such models have informed the study of development. Next, the chapter focuses on such comparisons. The chapter closes with a discussion of general issues in the information processing endeavor and promising directions for future work.

WHY MODEL?

At first blush, computational approaches seem to require more motivation and justification than perhaps all other prevalent approaches to the study of development, despite each approach having its unique strengths and limitations. Behavioral studies with children and adults are of obvious importance when the questions of interest concern how we behave and why, even though behavioral observations alone cannot specify the mechanisms underlying the behaviors. Neuroimaging approaches generate great excitement about the prospects of watching the brain as it thinks, despite providing only indirect measures of neural activity. Studies with special populations and patients with brain damage may shed light on not only atypical but also typical functioning, despite the complexity involved in trying to make inferences about exactly what mechanisms have been impaired, preserved, and compensated for in such cases. And animal studies can support invasive measures and controlled environments not possible in human studies, even if the lessons learned may not always generalize to other species. All these methods are generally appreciated in the context of their strengths and their limitations.

Computational models of development, and of cognition and behavior more generally, similarly have their strengths and limitations, but their potential may be less apparent. After all, watching an artificial system on a computer as it develops and behaves does not have quite the impact of observing the behaviors of real children or nonhuman animals, or their corresponding brain images. So why model? Many compelling arguments and examples have been put forth to answer this question (e.g., Elman et al., 1996; Klahr, 1995; O'Reilly & Munakata, 2000; Simon & Halford, 1995a). A common theme to these answers is that it is challenging to understand how change occurs, an issue at the heart of developmental study (Siegler, 1989). Flavell (1984) noted, "Serious theorizing about basic mechanisms of cognitive growth has actually never been a popular pastime... The reason is not hard to find: good theorizing about mechanisms is very, very hard to do" (p. 189). Computational models may provide a particularly useful tool for this challenging task.

One benefit of computational models may be most evident when considering other fields of study, such as meteorology and physics, where the need for models as a complementary approach is highly appreciated. Understanding the weather and the physical world requires comprehending complex interactions among many elements. Various phenomena emerge from such interactions, so that the whole is impossible to understand by considering the elements in isolation. As a simple example, consider two gears of different sizes that can interlock. To understand how they behave, it is insufficient to consider each in isolation. Instead, behavior emerges from the interaction of the two gears, with the smaller gear driving the larger gear to yield a decrease in rotational speed and an increase in torque. As the interactions among elements become more intricate, as in the weather and physics, computational models become increasingly important. Such models allow the observation and manipulation of interactions among elements and associated emergent phenomena. Similarly, models can be essential in helping us comprehend the intricacies of all the interacting elements that produce our thoughts and behaviors. And, understanding this complexity may be even more challenging during intense periods of change in thinking and behaving, as observed during childhood. Thus, one reason to use models in the study of development is the extraordinary complexity of the issues under question; development is far too complicated to be defined solely in terms of simple processes.
that can be captured through purely verbal description. As other methods become increasingly sophisticated, providing large amounts of detailed information about our behaviors and their neural bases, computational models will only become more important in helping to make sense of this complexity.

A second answer to the question of “Why model?” focuses on the need to be explicit about assumptions and constructs in theory development and evaluation. Purely verbal theories may rely on constructs that are not specified well enough to be rigorously tested or understood. The Piagetian notions of assimilation and accommodation were criticized in this regard:

For 40 years now we have had assimilation and accommodation, the mysterious and shadowy forces of equilibration, the Batman and Robin of the developmental processes. What are they? How do they do their thing? Why is it after all this time, we know no more about them than when they first sprang on the scene? What we need is a way to get beyond vague verbal statements of the nature of the developmental process (Klahr, 1982, p. 80).

Klahr (1995) also said of one of Piaget’s descriptions of assimilation and accommodation, “Although it has a certain poetic beauty, as a scientist, I do not understand it. I do not know how to test it, and I doubt that any two readers will interpret it in the same way.” Creating a working computational model of such processes forces one to address these issues explicitly and to confront aspects of the problem that might have otherwise been ignored. The resulting model can also be run to generate novel predictions. Thus, creating models can help make explicit theoretical assumptions, constructs, and predictions—an essential step in evaluating and advancing theory. Again, as other methods provide increasingly detailed information about brain and behavior, and various theories are formulated to make sense of such data, computational models will only become more important in assessing the viability of such theories.

A third reason to model is that computational models arguably allow the greatest levels of control for testing theories. A single variable, such as the firing rate of artificial neurons or exposure to particular words in the environment, can be manipulated in isolation to see the effects on the functioning and development of a simulated system. Multiple variables can be manipulated in a coordinated way to observe their interactions. This kind of control is essential for making headway on some of the thorniest questions in development. Debates about the roles of nature versus nurture, domain-general and domain-specific learning mechanisms, and so on, often hinge on what could (or could not) be learned from general learning mechanisms and exposure to the typical environment. Could infants come to understand the physical world through general learning mechanisms and exposure to objects in the world? Do children require innate, language-specific learning mechanisms to make sense of the language surrounding them? The remarkable degree of control afforded by models allows researchers to test the role of factors in such debates in ways that would otherwise be impossible. As other methods reveal more and more potential factors affecting thought and behavior, models will become increasingly important for allowing a controlled and systematic investigation of the possible effects of such factors.

Finally, models can also be helpful for providing a unified framework for understanding behavior (e.g., Anderson et al., 2004; Newell, 1990; Rumelhart & McClelland, 1986b). Such unified frameworks can support more stringent testing because they can be evaluated across a range of behaviors instead of a single phenomenon. Unified frameworks can also encourage more parsimonious explanations, rather than what sometimes seems like a hodgepodge of explanations proposed across development. Infants show a gradual progression in their understanding of object permanence, the continued existence of objects after they are no longer perceptible (Piaget, 1954). Infants first show a sensitivity to object permanence in their looking times to unexpected events with hidden objects (Baillargeon, 1999; Spelke, Breinlinger, Macomber, & Jacobson, 1992), then by reaching for objects hidden in the dark (Goubet & Clifton, 1998; Hood & Willatts, 1986; Shinskey & Munakata, 2003), followed by reaching for objects hidden in the light (Piaget, 1952b). Later still, infants succeed in reaching for an object hidden in a new location after it was repeatedly retrieved from a different hiding location (Diamond, 1985; Piaget, 1954). Some accounts attribute each of these task-dependent developments to a different factor, with motor developments supporting successful reaching in the dark; problem-solving developments supporting search in the light; and working memory and inhibitory developments supporting successful searching with multiple hiding locations (e.g., Baillargeon, Graber, DeVos, & Black, 1990; Diamond, 1985; Willatts, 1990). Although each of these independent factors may contribute to the observed developmental progression, more unified developmental processes may also play a role. Models provide a natural framework for exploring such possibilities.
For all these reasons, models provide an important tool for understanding the complexity of development. Models are an essential complement to other methods, and vice versa. And, the need for models should only increase as we learn more about behavior and the brain, and must then formulate and evaluate increasingly complex theories of development.

**HISTORICAL CONTEXT**

In many ways, the history of information processing approaches to development parallels the history of information processing approaches more generally. These approaches developed as part of the cognitive psychology movement that served as a contrast to behaviorism, they began with relatively rigid notions of cognitive structures and processing, and they became increasingly dynamic and emergent as the field progressed. The behaviorism movement, which focused on explaining behavior without reference to mental processes, dominated psychological work during much of the first half of the twentieth century (Skinner, 1953; Watson, 1912). In sharp contrast, cognitive psychologists in the 1950s began embracing questions about the nature of inner thought processes (Bruner, Goodnow, & Austin, 1956; Chomsky, 1957). Within the field of cognitive psychology, some early information processing theorists focused on the computational bases for thought processes.

A critical step in the development of information processing approaches was the idea that cognitive theories could be stated (and run) as computer programs (Newell & Simon, 1972). Many cognitive theories (e.g., Anderson & Lebiere, 1998; Newell & Simon, 1972) were developed around the computer metaphor, with human cognition viewed as similar to processing in a standard serial computer (e.g., with cognitive processing separated from knowledge, in the same way that the processing machinery of a computer's central processing unit [CPU] is separated from knowledge structures in random-access memory [RAM]). However, the idea of stating cognitive theories as computer programs does not actually require that human cognition resemble processing on a computer. In the same way, computer models of the weather do not require that weather forces resemble computer processing. Instead, models can be completely distinct from the computers on which they are implemented and tested.

In fact, other information processing frameworks, such as neural networks and dynamic systems, increased in prominence in part as challenges to some of the more rigid notions of cognitive processing associated with the computer metaphor. In neural network and dynamic system frameworks, the lines between knowledge and processing are relatively blurred. Much of cognitive processing occurs in parallel, rather than serially. And thinking unfolds in an emergent, dynamic way, through the interactions of many low-level processes. Over time, production systems have also incorporated some of these characteristics (e.g., Anderson, 1983).

Information processing approaches to development followed a similar chronology. With Piaget's (1952b, 1954) extensive observations and theorizing about children's thinking, the field of cognitive development was born. The relevance of information processing approaches for the study of development was noted early in their history:

If we can construct an information-processing system with rules of behavior that lead it to behave like the dynamic system we are trying to describe, then this system is a theory of the child at one stage of the development. Having described a particular stage by a program, we would then face the task of discovering what additional information-processing mechanisms are needed to simulate developmental change—the transition from one stage to the next. That is, we would need to discover how the system could modify its own structure. Thus, the theory would have two parts—a program to describe performance at a particular stage and a learning program governing the transitions from stage to stage (Simon, 1962, pp 154–155)

According to this view, developmental processes can be described in terms of programs that transform earlier stages into later ones, and the stages themselves can also be described in terms of a different set of programs. Many early production systems of development fit this view, in which programs for performance in a given stage were distinct from programs for transitions between stages (Baylor & Gascon, 1974; Klahr & Wallace, 1976; Young, 1976). Just as some rigid distinctions in cognitive processing were blurred over time within information processing approaches more generally, this distinction between performance and transition mechanisms in development has become more blurred as the field has progressed (Klahr & MacWhinney, 1998). In many recent information processing approaches, transition mechanisms operate throughout a model's development, and common processes contribute to a model's stable performance and transitions between stable periods.
The point about distinguishing between models and computers has not always been appreciated in the context of information processing models of development. Such approaches have been criticized because "A system that cannot grow, or show adaptive modification to a changing environment, is a strange metaphor for human thought processes which are constantly changing over the life span of an individual" (Brown, 1982). Because models and the computers they run on are distinct, this criticism can apply to computers without applying to the computational models running on them (Klahr & MacWhinney, 1998). This point should become clear in the context of the numerous adaptively modifying models described in this chapter.

Because the information processing umbrella is so broad, including a wide range of cognitive theories, many other historical developments were fairly specific to particular variants of information processing approaches. Advances in neurobiology and neural inspired models and theories (Hebb, 1949; McCulloch & Pitts, 1943; Rosenblatt, 1958; Shepherd, 1992) were particularly informative for the development of neural network approaches. Advances in the understanding of complex systems at the biological, mathematical, and psychological levels (Kuo, 1967; Lehrman, 1953; Lewin, 1936; von Bertalanffy, 1968; Waddington, 1957) laid the foundation for dynamic systems approaches.

Building on such foundations, and on the key idea that cognitive theories can be stated as computer programs, other major contributions in the history of information processing approaches took the form of introductory texts, which often included computational models for the reader to explore. Such introductory texts played a large role in widely disseminating ideas behind production systems (Anderson, 1976; Newell & Simon, 1972), neural network models (McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986b), and dynamic systems approaches (Kelso, 1995). Similarly, introductory texts arguably generated a great deal of interest and activity in investigating developmental questions through information processing approaches, with neural networks (Elman et al., 1996; Plunkett & Elman, 1997), dynamic systems (Thelen & Smith, 1994), and production systems (Klahr & Wallace, 1976).

OVERVIEW AND EXAMPLE OF EACH KIND

This section focuses on four major types of information processing approaches: production systems, neural networks, dynamic systems, and ad hoc models. Each type is first considered in isolation, in terms of the basic assumptions and components of the approach, and the kinds of applications to development that have been investigated, with one example considered in detail. At the end of this section, broad comparisons are drawn across some of these types of information processing approaches. The next section provides more detailed comparisons, by evaluating different models of the same developmental phenomena.

Production Systems

There are many variations of production systems, such as SOAR (Newell, 1990), ACT-R (Anderson & Lebiere, 1998), and 3CAPS (Just & Carpenter, 1992).

Basics

As described in many sources (e.g., Anderson, 1993; Klahr, Langley, & Neches, 1987; Klahr & MacWhinney, 1998; Newell & Simon, 1972), production systems focus on cognitive skills that take the form of production rules. Production systems consist of two interacting structures:

1. Production memory: This is the system's enduring knowledge, and consists of a large number of condition-action (or IF-THEN) rules called productions. The conditions specify the circumstances under which the production applies. In the case of number conservation (Piaget, 1952a), the condition for one production might be: "If you have a goal of stating the numerical relation between two collections, and the collections had the same number of objects before a transformation and the transformation did not involve adding or subtracting objects" (Klahr & Wallace, 1976). The actions specify the actions to be taken under those circumstances. The corresponding action for the number conservation example might be: "then the rows still have the same number of objects." The conditions and actions for productions can apply to either the external world or to mental states.

2. Working memory: This is the system's representation of the current situation and consists of a collection of symbol structures called working memory elements. Information in working memory may come from both the external world and through the actions associated with productions.
The organization of these structures is shown in Figure 10.1.

Distinct processes relate production memory and working memory as follows:

1. **Recognition or matching** process: This process finds productions with conditions that match information in working memory. This process can lead to conflict, because several productions may have conditions that match the current state of working memory, and a single production may match the current state of working memory in different ways.

2. **Conflict resolution** process: This process selects which of the matching productions will be applied.

3. **Act** process: This process applies the actions of the selected productions.

This process works iteratively, with each action leading to new information in working memory, so that the three-step cycle begins again. This processing has both parallel and serial components. The search for productions with conditions that match the contents of working memory occurs in parallel. The execution of actions occurs in serial.

Learning in these production systems takes the form of the creation and modification of productions as a result of experience. For example, redundant steps may be dropped from productions, and two productions may be combined into one, as a result of experience.

**Applications to Development**

Production systems have been used for understanding many different aspects of cognitive development, including higher level cognitive processes such as problem solving across different domains (e.g., Klahr & Siegler, 1978; Klahr & Wallace, 1976; Simon & Halford, 1995a; van Rijn, van Someren, & van der Maas, 2003). This kind of tradition is evident in some more recent production systems models. Jones, Ritter, and Wood (2000) used production systems to investigate different theories of what leads to developmental improvements on a problem-solving task that requires constructing a pyramid from 21 wooden blocks. Production systems may provide the most natural fit for modeling such processes, given the way in which higher level information can be represented in production memory. However, as discussed in subsequent sections, other models can be applied to the study of such higher level processes as well. And as discussed next, production systems can be applied to the study of arguably more basic forms of processing.

One example concerns infants’ understanding of number. Can infants add and subtract? Do they “innately possess the capacity to perform simple arithmetical calculations” (Wynn, 1992)? Some researchers have argued that the answer is yes, based on infants’ looking times to events that lead to expected or unexpected outcomes. In one event, infants see one toy sitting on a puppet stage. A screen is then raised, occluding the toy. Infants then see another toy being placed behind the screen. The screen is then dropped, revealing either two toys (an expected event) or one toy (an unexpected event). Five-month-old infants look longer at the unexpected event (Wynn, 1992). Control conditions demonstrate that infants do not simply prefer to look at outcomes with one toy over outcomes with two toys. Thus, one interpretation is that infants look longer at the unexpected outcome based on their computation of $1 + 1 = 2$; after computing that two toys should be on the stage, they look longer at outcomes that violate this expectation.

However, this interpretation is controversial. Many alternative explanations have been proposed for infants’ longer looking times to unexpected outcomes. Infants might respond based not on the number of objects on the stage, but on some other factor that varies along with number, such as surface area (Feigenson, Carey, &
Spelke, 2002) or contour length (Clearfield & Mix, 1999; Mix, Huttenlocher, & Levine, 2002). Alternatively, infants might accurately track the objects without performing arithmetic computations (Simon, 1997). Or, infants might look longer at unexpected outcomes due to artifacts of the habituation procedures used in such studies, with infants responding on the basis of which displays are more familiar (L. B. Cohen & Marks, 2002).

Information processing approaches and computational models may be particularly useful in such cases of controversy. Models can help to clarify the assumptions of each theory, and exactly how they explain observed behaviors. Models of alternative accounts can serve as existence proofs that observed behaviors can arise through means other than those proposed. Such models can also be probed and manipulated to assess whether competing accounts represent true alternatives, or whether at core they are relying on common processes.

A production system model was put forth as an implementation of a “nonnumerical” account of infant behavior in violation-of-expectation studies (Simon, 1998). This model simulated infants’ longer looking to unexpected outcomes, based on domain-general processes: memory, individuation, object permanence, and spatiotemporal representations. Thus, the model was put forth as an existence proof of how infants’ responses can be understood without attributing numerical representations to them.

The model relied on creating indexes for each object encountered: physical object indexes for visible objects, and memory object indexes for objects that become hidden. The memory object indexes copied the spatiotemporal characteristics of the physical object indexes, and supported the formation of predictions. When such predictions were violated, longer looking times resulted due to searches for other possible matches in the display. The productions supporting this process are described in Table 10.1.

This production system model simulated infants’ looking times to standard “addition” (1 + 1 = 2) and “subtraction” (2 − 1 = 1) events. The workings of the model, which can be closely inspected, thus suggest the following possibility (Simon, 1997). Infants can encode spatiotemporal information about objects, use such information to individuate different objects, represent the continued existence of objects when hidden, and compare what they remember to what they see. These processes can support the recognition of an unexpected event, without requiring numerical representations or calculations. Thus, a “1 + 1 = 1” event can be recognized as unexpected simply because an object and another object are expected, but the display reveals only an object. In the same way, the sudden disappearance of an object can be recognized as unexpected, simply because an object was expected, and not because 1 − 0 = 1.

The working production system model allows various questions to be asked of this nonnumerical account. Is a nonnumerical account sufficient to explain the range of behaviors observed in infants, and could these be simulated within a single model? Would Simon’s (1997) model be able to simulate infants’ sensitivity to the ordinal relations of numbers (Brannon, 2002)? These kinds of questions highlight strengths of information processing approaches and computational models; the explicitness of assumptions can lead to a honing of theoretical claims (e.g., what it means for processes to be numerical or nonnumerical) and testing of behavioral predictions.

### Table 10.1 Productions

<table>
<thead>
<tr>
<th>Description</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longer looking to unexpected events is driven by the final production, which requires extra actions of checking additional locations for information when expectations are violated:</td>
<td>P1: If you have newly created physical object indexes and preexisting memory objects, then set a goal of comparing the predicted and actual states of the world.</td>
</tr>
<tr>
<td></td>
<td>P2: If your goal is to compare the predicted and actual states of the world, then set a goal of carrying out a one-to-one match process for each prediction.</td>
</tr>
<tr>
<td></td>
<td>P3: If your goal is to compare the predicted and actual states of the world, and you know the result of that comparison, then terminate looking.</td>
</tr>
<tr>
<td></td>
<td>P4: If your goal is to carry out a one-to-one match process for a prediction, then set a goal of seeing if a preexisting memory object matches a physical object.</td>
</tr>
<tr>
<td></td>
<td>P5: If your goal is to see if a preexisting memory object matches a physical object, and a match occurs, then verify the prediction for that preexisting memory object and move on to predictions for other preexisting memory objects.</td>
</tr>
<tr>
<td></td>
<td>P6: If your goal is to see if a preexisting memory object matches a physical object, and a match does not occur, then search other locations for other possible physical object matches.</td>
</tr>
</tbody>
</table>

Adapted from Simon’s (1998) nonnumerical production system model of infant behavior in violation-of-expectation studies.

**Neural Networks**

Neural network models are also known as connectionist or parallel distributed processing models. Each of these
names captures aspects of how processing occurs in these models, in parallel across a network of interconnected nodes.

**Basics**

There are many variations of neural network models (e.g., Arbib, 2002), but all models share the common features of units and weights (McClelland & Rumelhart, 1986; O'Reilly & Munakata, 2000; Rumelhart & McClelland, 1986b):

- **Units** are neuron-like entities that represent information through their activity, which can be communicated to other units.
- **Weights**, or connections, link units to one another. The strength of connections changes through learning, in a manner akin to changes in the efficacy of synapses with learning.

Information is represented in neural network models in terms of patterns of activation across units. Units are typically organized into layers, such as an input layer that represents information available in the environment, an output layer that represents the network's actions or decisions, and hidden layers in between that allow patterns of activity to be transformed between the input and output layers. The activity of any given unit is a function of the activity of other units and the strength of the connections between units.

Representations of information in neural networks are often distributed, graded, and interactive. For example, a network's representation of the concept of number might be distributed across multiple units in the network, which also participate in the representation of related concepts (e.g., quantity and counting). These units can vary in their levels of activation, supporting gradedness in the strength of the associated concepts. And, because of the connections that send activity from one unit to another, networks of units can be highly interconnected, leading to interactive representations that can have large effects on one another.

Learning occurs through changes to connections between units. A number of learning algorithms have been investigated through neural network models. Two broad classes can be delineated (O'Reilly & Munakata, 2000): error-driven and self-organizing learning. Error-driven learning is guided by the goal of reducing errors, measured as a function of the difference between target activations and a network's actual activations. Such targets can come from various sources, such as a teacher explicitly correcting a student's behavior, the environment providing a target signal for one's expectations about what will happen next, and a person's goal for a motor action providing a target signal for the attempted motor action. Backpropagation (Rumelhart, Hinton, & Williams, 1986a) is one of the most common variants of error-driven learning algorithms.

Self-organizing learning entails forming representations that capture important aspects of the environmental structure, based on patterns of simultaneous activation among processing units. Hebbian learning algorithms (e.g., Oja, 1982), whereby "units that fire together, wire together," are one of the most common forms of self-organizing learning algorithms. Algorithms have also been developed that combine error-driven and self-organizing learning, and demonstrate how each benefits from the other (O'Reilly & Munakata, 2000).

**Applications to Development**

Neural network models have been applied to a broad range of domains in the study of cognitive development (see reviews in Elman et al., 1996; Munakata & McClelland, 2003; Quinlan, 2003; Shultz, 2003), including language (Elman, 1993; Plunkett & Marchman, 1993; Seidenberg & McClelland, 1989), categorization (Mareschal & French, 2000; Rogers & McClelland, 2004), and object knowledge (Mareschal, Plunkett, & Harris, 1999; Munakata, McClelland, Johnson, & Siegler, 1997). Many of these models have been focused on understanding the kinds of learning processes and representational changes that support the changes observed in infants and children during development. Most of these models have focused on typical development. However, as discussed next, neural network models can also be informative in the study of atypical development, given their abilities to capture nonlinear dynamics and emergent properties of complex developing systems (Morton & Munakata, 2005).

In the study of developmental disorders, the nature-nurture debate takes a familiar form. Early arguments pitted genetic contributions against environmental ones, whereas now researchers generally agree that both contribute in important ways. Nonetheless, two contrasting perspectives can still be identified (Karmiloff-Smith,
2005). A modular approach assumes (implicitly or explicitly) a static view of brain function, in which neural systems or modules are innately specified for particular functions. From this perspective, developmental disorders arise due to genetic alterations that target particular associated cognitive functions. As a result, a developmental disorder may lead to specific cognitive impairments, similar to those observed in cases of adult brain damage. In contrast, within the neural network framework, it is more natural to consider particular functions emerging in particular brain areas through a highly interactive process of development. Rather than different brain regions simply having their functions prespecified, they develop as they do in part because of how other regions are developing and through small differences in start state. From this perspective, developmental disorders emerge through genetic alterations leading to small changes in low level properties of a system’s start state that interact with processes of development (Karmiloff-Smith, 1998).

Information processing approaches and computational models may be particularly useful for investigating the complexity of such emergent processes. A number of neural network models have demonstrated how small, quantitative differences in the starting state of systems can lead through a process of development to qualitative differences in outcome (Harm & Seidenberg, 1999; Joanisse & Seidenberg, 2003; Oliver, Johnson, Karmiloff-Smith, & Pennington, 2000; O’Reilly & McCleland, 1992; Thomas & Karmiloff-Smith, 2002, 2003). Because of the complexity of the developmental process, damage to a system early in development can lead to very different behaviors than damage to the same system late in development (Thomas & Karmiloff-Smith, 2002, 2003).

One model demonstrated how deficits in syntactic processing, observed in individuals with specific language impairment (SLI), can arise through development from small disturbances in phonological representations (Joanisse & Seidenberg, 2003). The model simulated the sentence comprehension profile of individuals with SLI across sentences of varying complexity. Thus, the model demonstrated how apparently specific deficits can arise through low-level variations, rather than the genetic specification of cognitive modules.

The model’s task was to map sequences of words onto their meanings. The model consisted of four layers: a phonological input layer, two hidden layers, and a semantic output layer. The hidden layers sent and received activity from one another, which allowed information to be maintained across several time steps. These layers thus supported the network’s working memory, which could aid in the mapping of a word onto its meaning. To explore the effects of small differences in start state, two versions of this model were trained: an intact version, and a version in which the phonological inputs to the network were distorted through the addition of a small amount of noise.

These networks were trained on a corpus of 40,000 sentences that varied in syntactic complexity. Sentences were presented one word at a time, and the networks had to identify the meaning of and syntactic dependencies between each word to activate an appropriate set of units in the semantic output layer. For example, the sentence “John says Bill likes himself” was presented to the network as “John” “says” “Bill” “likes” and “himself.” After the presentation of the final word “himself,” the network was trained to activate the meaning of this word in this context: the [MALE], [REFLEXIVE-PRONOUN], [HUMAN], and [BILL] units of the semantic output layer. After training, syntactic processing was tested using novel sentences that were not part of the training corpus. The test set included sentences with reflexives that could only be resolved syntactically (e.g. “Harry says Bob likes himself”), and similar sentences but with nouns or reflexives modified so that the reflexive could be resolved with the help of gender information (e.g. “Sally says Bob likes himself”).

The intact network performed near ceiling on both sets of sentences, simulating the performance of typical individuals. In contrast, the network with phonological deficits showed a selective deficit in resolving reflexives based on syntactic information alone, simulating the performance of individuals with SLI. Thus, a low-level deficit in phonological processing led to selective grammatical deficits. Specifically, the deficit in phonological processing made it more difficult for the network to form and maintain consistent representations in working memory. For sentences such as “Harry says Bob likes himself,” in which the meaning of “himself” could only be resolved on the basis of syntactic information, the network’s working memory representations were insufficient to resolve the sentences. In contrast, for sentences such as “Sally says Bob likes himself,” which could be resolved on the basis of syntactic and semantic information, the network’s working memory
representations were more robust to noise, so that these sentences could still be resolved.

Dynamic Systems

Dynamic systems approaches focus on understanding complex, nonlinear changes in systems over time, through both verbal description and formal simulation.

Basics

There are many variations in this approach (e.g., von Bertalanffy, 1968; Fischer & Bidell, 1998; Kelso, 1995; Smith & Thelen, 1993; van der Maas & Molenaar, 1992), but most include some form of the following ideas:

- Behavior is multiply determined, influenced by processes at multiple levels, within the organism and between the organism and the environment.
- Systems are softly assembled (Kugler & Turvey, 1987), flexibly adapted in a self-organizing manner, rather than being hard/wired or programmed.
- Some resulting states are more stable than others, and constitute "attractors."

In the case of walking, movements are viewed as softly assembled based on factors such as the environment, level of arousal, leg mass, and so on (Thelen & Ulrich, 1991). Certain patterns are more stable than others (e.g., walking, trotting, and galloping in the case of quadrupeds). This kind of dynamic systems approach contrasts with the notion of a central pattern generator that endogenously generates the neural activation patterns that drive locomotion. Within the dynamic systems framework, one goal is to understand the processes at multiple levels that contribute to behavior, and how different parameter values of those processes lead to different attractors and behaviors.

Applications to Development

Dynamic systems approaches have been adopted to investigate aspects of development (Lewis, 2000; Thelen & Smith, 1994, 1998) from early work on personality development (Lewin, 1935, 1936), to seminal work on motor development (Thelen, Kelso, & Fogel, 1987), to more recent work in domains that might be considered more cognitive (Thelen, Schöner, Scheier, & Smith, 2001; van Geert, 1998). Much of the work in this tradition has taken the form of verbal theories, using dynamic systems constructs as a metaphor for reconceptualizing development (Spencer & Schöner, 2003). This is a useful and important step, but implemented models should aid in the assessment of this approach and comparison to alternatives.

One set of implemented simulations focused on cognitive growth (van Geert, 1991, 1993, 1998). The overall approach with these models was to identify a set of relevant variables in the domain of interest, express the relations between these variables in mathematical equations, assign parameter values in the equations, and then run the equations to test the fit of the model's developmental trajectory to the observed behavioral data. When good fits were observed, the models served as a demonstration that development might proceed as hypothesized, in terms of the posited variables, relations, and parameter values.

A specific model focused on lexical and syntactic growth in children (van Geert, 1993). In this case, the relevant variables might include things like the number of words and different syntactic rules acquired at a given point in time. Various parameters to be set in the equations included growth rate (the amount of growth over a specific time interval), and feedback delay (which affected how quickly a given state of the system affected subsequent states). One curve to be fit characterized the number of words in a child's lexicon week by week (Dromi, 1986). Poor fits to the empirical data were obtained with no feedback delay; better fits were obtained with a feedback delay of 2 weeks and growth rate of .71 for the early part of the empirical curve, and with a feedback delay of 1 week and growth rate of .35 for the later part of the empirical curve. Models of this sort have also been applied to fitting curves of various general aspects of development, including discontinuous change (van Geert, 1998).

Clear strengths and weaknesses have been identified with this type of dynamic systems model (Aslin, 1993; Thelen & Smith, 1998). One strength is forcing experimenters to think about their systems in precise ways, by collecting detailed data that are amenable to dynamic models and carefully considering potential contributing components, their relations, and relevant parameter values. Like all models, these models can also lead to empirical predictions that can be tested. However, some examples of this kind of modeling approach have been criticized for being "theory-rich and data-poor" (Thelen & Smith, 1998), with much of the models' components, interactions, and parameters being largely hypothetical.
As a result, the process of curve-fitting may be too unconstrained to be informative (Aslin, 1993).

### Ad Hoc Models

Ad hoc models refer to models focused on the information processing demands of the domain under consideration, without being constrained by all the assumptions and claims of global frameworks, such as production system, neural network, and dynamic systems approaches (Klahr & MacWhinney, 1998). Ad hoc models can employ a wide variety of architectures and learning algorithms. For this reason, they cannot readily be compared as a group against other frameworks, and so will not be focused on beyond this section.

Ad hoc models have proven useful in the investigation of strategic development (e.g., Shrager & Siegler, 1998; Siegler & Shipley, 1995; Siegler & Shrager, 1984). Such models have demonstrated how relatively simple processes can support the discovery of new strategies and the making of adaptive choices among strategies—two abilities that children show very early (e.g., Adolph, 1997) and reliably (Siegler, 1996).

A unified model focused on choosing among strategies and developing new strategies; these processes had previously only been simulated in isolation across separate models (R. Jones & VanLehn, 1991; Neches, 1987; Siegler & Shipley, 1995; Siegler & Shrager, 1984). The unified model was called SCADS, for Strategy Choice and Discover Simulation (Shrager & Siegler, 1998). This model focused on the development of simple addition strategies and captured many aspects of children's behavior on addition problems, including choosing adaptively among diverse strategies and discovering the same strategies in the same sequence and manner as children do.

The model comprised three broad components:

1. An associative learning strategy choice component that represented strategies as sets of operators (e.g., "choose addend, say addend, clear echo buffer, count out fingers to represent addend") and recorded statistics about the speed and accuracy of strategies.

2. A working memory system that maintained traces of each strategy's execution and results, so that they were available for analysis.

3. A metacognitive system that analyzed strategies, identified potential improvements, and generated new strategies by recombining operators from existing strategies. This metacognitive system had three components: an attentional spotlight that increased resources allocated to new strategies, strategy-change heuristics that eliminated redundancy and recognized the importance of order of operations in strategies, and goal-sketch filters that prevented the execution of invalid strategies.

The model began each run with only two strategies: retrieval (providing an answer based on memory) and the sum strategy (put up the number of fingers corresponding to one addend, count, put up the number of fingers corresponding to the other addend, count, and finally count all the fingers up). The sum strategy was chosen often initially, because the system did not have strong enough associations between problems and answers to use retrieval. Attentional resources were needed in this early use of the sum strategy. With continued practice, attentional resources were freed so that the system could discover new strategies. If the strategies passed the goal sketch filters (which required both addends to be represented and used in reporting a result), the strategies were kept. If the strategies were efficient and accurate, they were used more and more. New strategies were also discovered through the elimination of redundant steps in existing strategies. The shortcut sum strategy (put up fingers for both addends and count them once) was formed from the sum strategy, by dropping the redundant steps of first counting the fingers for each addend. Next, the redundancy of counting the first addend could be dropped to simply state the value of that addend and count from there. This strategy was more efficient when starting with the larger addend, leading to the min strategy (counting up from the larger addend).

From these basic processes, the model chose adaptively among diverse strategies and discovered the same strategies in the same sequence and manner as children. The adaptive choice reflected the model's associative learning of the strengths of various strategies, based on their associated speeds and accuracies. The model's strategy discovery process was based on fairly simple constraints: satisfying the basic goal sketch, eliminating redundancy, and attending to the importance of order of operations. These constraints were sufficient to yield the same strategies that children use. Moreover, because these processes could occur whenever attentional resources were available, the model discovered new strategies in a manner parallelizing that of children; for
example, following correct as well as incorrect performance, and without requiring trial-and-error learning. The model discovered new strategies in the same order as children, in part through dropping redundant steps from one strategy to create a new strategy.

In this way, the SCADS model provided insight into the possible processes underlying children’s strategy discovery and choice. What might otherwise seem like mysterious processes may be driven by relatively basic processes of associative learning and heuristic knowledge use.

**General Comparisons across Frameworks**

Comparisons among information processing approaches are most straightforward in the context of different models of the same phenomenon, which is the focus of the next section. This section first briefly considers some general comparisons that may be drawn across production systems, neural network, and dynamic systems approaches. This is a difficult exercise, given the many varieties of each type of information processing approach. It is no coincidence that others who have attempted such comparisons have introduced them as “open to extensive criticism” (Thelen & Bates, 2003) or noted that others are likely to question and argue with the comparisons (Anderson & Lebiere, 2003). The same applies to the current comparison, summarized in Table 10.2. Dynamic systems approaches could be viewed as an overarching theory that includes dynamic models of all kinds, including neural network models as one particular case. And, for each characterization listed for each approach, exceptions can probably be found. But these characterizations capture some general differences between these approaches as they have typically been investigated.

**Representations**

Historically, these three approaches have been quite different in their treatment of mental representations. Production systems focused on a symbolic level of representations, with working memory consisting of symbol structures and production memory taking the form of If-Then propositions. Neural network models focused on a subsymbolic level of representations, the “microstructure of cognition” (Rumelhart & McClelland, 1986b) with representations distributed across connections and patterns of activation. Dynamic systems researchers largely eschewed the “R-word,” because of its possible connotation of a static entity sitting in the head (see discussion in Spencer & Schöner, 2003; Thelen & Bates, 2003).

Again, none of these characterizations are absolute. Production systems have incorporated continuous and noisy activation values (e.g., Anderson & Lebiere, 1998; Just & Carpenter, 1992), and neural network models have been used to address the development of symbol-like rules (Rougier, Noelle, Braver, Cohen, & O’Reilly, 2005). And some recent dynamic systems models have investigated representations, at a subsymbolic level, in terms of particular states of a system at particular times (Spencer & Schoner, 2003).

**Relative Strengths and Weaknesses**

Each approach has its relative strengths and weaknesses, which in some cases may be viewed as trade-offs, where one feature is emphasized at the expense of another. Production systems may have a relative strength in simulating flexible behavior because complex behaviors can be handled through sequences of productions (e.g., Taatgen, 2002), and symbolic representations can be used flexibly across variations in perceptual input. However, the flexibility of these models might come at the cost of being less suited to capture emergent effects in cognition. Small changes in perceptual processing can lead to large changes at the cognitive level, as explored in the study of developmental

<table>
<thead>
<tr>
<th><strong>Table 10.2 Comparison of Three Major Information Processing Frameworks</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Production Systems</strong></td>
</tr>
<tr>
<td><strong>Representations</strong></td>
</tr>
<tr>
<td><strong>Relative strength</strong></td>
</tr>
<tr>
<td><strong>Relative weakness</strong></td>
</tr>
<tr>
<td><strong>Strongest track record</strong></td>
</tr>
</tbody>
</table>

Note: These characterizations are not absolute, and exceptions exist, as described in the text.
disorders. These emergent effects may be better captured by nonproduction systems models, in which knowledge is represented in ways that are less flexible and more tied to specific lower level processes. Neural networks may have a relative strength in biological plausibility, in terms of the mapping between the computational elements of units and weights and biological elements of neurons and synapses, and in the development of biologically plausible learning algorithms and architectures (e.g., O'Reilly & Rudy, 2001). However, because of the way that these models learn based on particular experiences, with units becoming committed to representing specific information, the knowledge in such models has been criticized for not being sufficiently abstract (Marcus, 1998). Dynamic systems models may have a relative strength in their focus on embodiment, and the effects on cognition and development of being situated in a physical body. However, development in these models is often simulated through changes to external control variables, without explanation of how such changes come about. As a result, such models have not addressed learning processes that might underlie such changes.

Again, these characterizations are relative. Production systems have yielded predictions about neural function that have been tested through neuroimaging studies (Fincham, VanVeen, Carter, Stenger, & Anderson, 2002). Neural network models have been used to address questions about processes of abstraction and generalization (Christiansen & Curtin, 1999; Munakata & O'Reilly, 2003; Rougier et al., 2005; Seidenberg & Elman, 1999). And learning has been incorporated into some dynamic systems models (van Geert, 1993).

**Strongest Track Record**

Finally, these three approaches have differed in their domains of application. Production systems approaches arguably have their strongest track record in simulating aspects of higher-level cognition, such as problem solving. Neural network models have addressed many aspects of language, whereas dynamic systems approaches have probably had their greatest impact in the study of motor processing. Such distinctions may be based in part on how readily different modeling approaches can fit the empirical phenomena. However, these approaches also have some overlap in their domains of application, or the next section of this chapter would not be possible.

**DETAILED COMPARISONS: DIFFERENT MODELS OF THE SAME PHENOMENA**

This section compares different types of models that have been applied to understanding the same developmental phenomena, across the domains of problem-solving, language, and memory. In the problem-solving domain, children's development on the Piagetian balance scale task (Inhelder & Piaget, 1958) has been investigated through production systems and neural network approaches. In the domain of language, children's developing abilities to conjugate the past tense of verbs has been investigated through neural network and production systems approaches. Finally, in the domain of memory, infants' memory for hidden objects as assessed through the Piagetian A-not-B task (Piaget, 1954) has been investigated through neural network and dynamic systems approaches. In each case, the insights afforded by each modeling approach are compared and contrasted.

**Problem Solving: The Balance Scale Task**

Children appear to pass through qualitatively different stages in solving certain tasks (Case, 1985; Piaget, 1952b). In the balance scale task (Inhelder & Piaget, 1958), children view a scale with weights on each side at particular distances from the fulcrum, and they must decide which arm of the scale will fall when supports underneath the scale are released. Children initially answer problems randomly, using no apparent rule about the physical properties of weight and distance to guide their decisions. They then employ different information to help them solve the task, progressing through four rules (Figure 10.2, Siegler, 1976, 1981). With Rule I, children attend to only the amount of weight on each side of the fulcrum. With Rule II, children also attend to the distance of weights from the fulcrum, if weights are equal on each side of the fulcrum. With Rule III, children consider both weight and distance information in all cases, but when this information conflicts, children make a random prediction. Finally, Rule IV represents mature knowledge of the task, which requires the computation of torque (sum of weights times distances).

Models of many different flavors have been applied to understanding children's performance on the balance scale task (Newell, 1990; Sage & Langley, 1983; Schmidt & Ling, 1996; Shultz, Schmidt, Buckingham, & Mareschal, 1995). For purposes of comparison,
the current discussion focuses on an early production system model (Klahr & Siegler, 1978), a subsequent neural network model (McClelland, 1989, 1995), and a recent production system model (van Rijn et al., 2003).

**Early Production System Model**

The early production system model of balance scale performance (Klahr & Siegler, 1978) focused on the productions and operators required to carry out the four rules used by children on the balance scale task. Thus, this model provided a more precise characterization of the dynamics of processing underlying the decision tree representation shown in Figure 10.2.

The productions for models employing each of the four rules are listed in Figure 10.3. For each production, the conditions are shown on the left and the actions are shown on the right. The conditions all check for same-ness or difference, in weight, distance, or torque. Where the conditions for a production match the information in working memory, the associated action (saying which side will go down) is produced. In the simplest model (employing Rule I), if the information in working memory indicates that the two sides have the same weight, P1 will fire and the response will be to say that the two sides balance. If the information in working memory indicates that one side has more weight than the other, P2 will fire and the response will be to say that side will go down.

The models become more elaborated to capture more advanced rule use (Figure 10.3). The model employing Rule II has an additional production, whereby if the two sides have the same amount of weight but one side has more distance, the response will be to say that side will go down. Based on this specification of what is required to use each rule, it is straightforward to see what modifications are required to progress from one model to the next (bottom of Figure 10.3). Transitioning from Rule I to Rule II requires adding the production that responds based on unequal distance when the weights are equal, and adding operators that encode and compare distances. This model did not actually simulate such transitions however.

**Neural Network Model**

Subsequent models have focused more attention on the transitions between rules. A neural network model of the balance scale task (McClelland, 1989, 1995) demonstrated how stage-like progressions from one rule to the next can result from small, successive adjustments to connection weights.

The model (Figure 10.4a) consisted of an input layer representing weights and distances on the left and right sides of a balance scale, a hidden layer with separate units for representing weight and distance information, and an output layer representing the choices for which side should go down: left, right, or balance (if the activations of the two output units were similar). The input layer represented weight and distance information in terms of localist patterns of activation for each side of the balance scale, with each unit corresponding to a certain number of weights or a certain distance from the fulcrum. In the problem shown in Figure 10.4, there are four weights located two pegs from the fulcrum on the right side. These are represented through activation of the corresponding units in the input layer (the 4th weight unit and the 2nd distance unit for the right side).

This model was presented with many balance scale problems of this sort. It received greater exposure to

---

**Figure 10.3** Production system representations of children's rules on the balance scale task (Klahr & Siegler, 1978). W = Weight and D = Distance. Source: From “Information Processing” (pp. 631–678), by D Klahr and B MacWhinney, in *Handbook of Child Psychology*, volume 2, fifth edition, W Damon (Editor-in-Chief), and D Kuhn and R S Siegler (Eds.), 1998, New York: Wiley. Reprinted with permission.
According to the backpropagation learning algorithm: The model activated responses to each problem that was presented to it (which could be viewed as predictions about balance scale outcomes), and its connections were adjusted to reduce errors—the discrepancy between the model’s output and the actual outcome. Based on this experience, the model progressed from random responding to Rule I behavior, from Rule I to Rule II behavior, and from Rule II to Rule III behavior. This progression was stage-like, in that the model showed relatively stable performance on each rule, punctuated by relatively rapid transitions between rules.

Why did the model display stage-like transitions? The network’s initially random weights were slowly modified with each experience to reduce the discrepancies between the network’s predictions about balance scale problems and the actual outcomes. The network first began to develop representations of weight in its hidden layer, given the greater predictive power of this factor in the problems presented. Early in this process, the network’s output still reflected random answers because the connections from between the input, hidden, and output layers were not yet sufficiently meaningful. As these connections became more fully formed, units became more distinct in their activation patterns, such that changes to connections could proceed rapidly to stage-like improvements in the network’s performance. Figure 10.4b shows these accelerations in the connection weight changes for the connections from the input to the hidden layer (bottom) and from the hidden layer to the output layer (top). The accelerations in the connections for the weight information correspond to the transition to the Rule I stage, and the accelerations in the onset of the distance information correspond to the transitions to Rules II and III. In this way, incremental weight adjustments in neural networks can result in small representational changes that then support relatively fast learning, producing stage-like behavior.

Although the model simulated children’s stage-like progression from no-rule behavior to Rule I, Rule II, and Rule III behaviors, it did not capture a clear transition to the most sophisticated Rule IV; instead, at the end of training, the model vacillated between Rules III and IV. This might reflect that a different kind of learning (e.g., based on explicit teaching) may underlie the transition to Rule IV use (McClelland, 1995).

**Recent Production System Model**

A recent production system model of the balance scale task (van Rijn et al., 2003) was aimed at addressing...
several possible limitations with earlier models. Although McClelland's neural network model relied on learning from errors to drive developmental transitions, children can show transitions in their balance scale performance in the absence of feedback. A small percentage of children switched from Rule I to Rule II after being presented with problems that highlighted distance information, without any feedback (Jansen & van der Maas, 2001). These problems involved gradually and systematically increasing the distance between weights and the fulcrum across problems, and then decreasing this distance. In this sequence, 4% of the children switched from Rule I to Rule II and back again at the same point in the sequence (as distance increased and as it decreased); 3% switched from Rule I to Rule II and back again at different points in the sequence; and 9% switched from Rule I to Rule II as the distance difference increased and did not switch back to Rule I. To capture such effects, a model would need to learn not only in response to error feedback.

In addition, the production system model was aimed at capturing qualitative transitions between rules. Although McClelland's model was characterized as capturing stage-like progressions based on quantitative underlying changes, it has been criticized for not being truly stage-like. This model has been criticized for not showing qualitative transitions between rules (Raijmakers, van Koten, & Molemaa, 1996) and for not following a distinct set of rules as analyzed through a statistical technique of latent class analysis (Jansen & van der Maas, 1997).

Van Rijn et al.'s (2003) production system model was aimed at capturing these kinds of phenomena, as well as the basic developmental progression observed from Rule I to Rule IV. The model was composed of IF-THEN productions and knowledge in the form of declarative memory chunks. There were three main factors underlying the model's behavior and development:

1. **Mechanisms**: These included processes such as the composition of new production rules, and the updating of the values associated with such productions (their utility), as well as with declarative memory chunks (their activation levels). Another mechanism was the general strategy of solving balance scale problems by searching for differences between the left and right sides of the balance scale.

2. **Task-specific concepts (weight, distance, addition, and multiplication)**: These were represented as chunks in declarative memory, with their availability mediated by the activation of these chunks.

3. **Capacity constraints**: These limited the number of differences the model could search for in trying to solve balance scale problems.

The concepts and capacity constraints were manipulated in the model to simulate differences across development (Figure 10.5). The task-specific concepts (weight, distance, etc.) were made available to the model at different points in its development, through changes to the activation of the declarative chunks representing those concepts. The capacity limitations were manipulated so that early in development models could search for only one difference between the two sides of the balance scale, whereas later in development, they could search for more than one difference. The concepts and capacity manipulations were motivated by empirical observations about children's encoding (Siegl, 1976; Siegl & Chen, 1998) and developmental theories about capacity (Case, 1985).

The model simulated the progression from Rule I to Rule IV and showed stable performance with each rule. Before any of the task-specific concepts were sufficiently activated, the model could only generate answers by guessing, which led to poor performance and low utility associated with this strategy. As soon as the

![Figure 10.5 Availability of components in the van Rijn et al. (2003) production system model of the balance scale task Source: From "Modeling Developmental Transitions on the Balance Scale Task," by H van Rijn, M van Someren, and H., van der Maas, 2003, Cognitive Science, 27. pp 227-257. Reprinted with permission.](image-url)
weight concept became available, the model thus began to use this concept (Rule I). When the distance concept became available, the capacity constraints limited the model to attending to only one dimension at a time. This led to the model attending to distance only if the weights were equal (Rule II). When the model's capacity increased to include more than one dimension, the model then considered weight and distance (Rule III). Finally, when the concept of multiplication became available, the model could then progress to computing torque (Rule IV). Behaviors associated with each of the rules were stable, based on the availability (or lack) of task-specific concepts and capacity, and the utility associated with production rules.

The model also simulated the finding that transitions from Rule I to Rule II may occur in the absence of feedback, and it showed the three types of transition patterns observed behaviorally (Jansen & van der Maas, 2002). As in the behavioral studies, the model was presented with sequences of problems with increasing and then decreasing distance differences, without feedback. The activation formula was revised to include a saliency term, computed as a function of the distance difference. As a result, problems with bigger distance differences were more salient, and led to increased activation of the distance concept. As soon as the distance concept became sufficiently activated, the distance values were used in solving the problem. That is, the model transitioned from Rule I to Rule II. If this occurred while the distance difference was increasing across problems, distance would continue to be used because the concept's activation would be greater from both the previous problem and the increased salience of distance information on the current problem. When the distance differences began to decrease across problems, the distance concept's total activation also began to decrease. Differences between models (e.g., in activation updating) contributed to whether or not the distance concept activation became so low that the model reverted back to Rule I.

Comparison of Models

A distinct difference among the models of the balance scale previously discussed is whether they attempted to account for the transitions between rules. The earliest model (Klahr & Siegler, 1978) did not, while subsequent models (McClelland, 1989, 1995; van Rijn et al., 2003) did. The fact that both of the subsequent models attempted to account for transitions demonstrates that this is not a distinction between neural network and production system models. However, these models attempted to address transitions in notably different ways. The neural network model explained transitions in terms of the same kinds of learning mechanisms applied in a consistent way across experience, with stages arising through changes in representations and how they could be used. In contrast, the production system model explained transitions through the introduction of new knowledge and capacity. On the one hand, these could be viewed as conflicting explanations; the production system model posited changes that are viewed as unnecessary within the neural network. On the other hand, these could be viewed as explanations at different levels; perhaps changes in the production system model could be implemented through changes at the neural network level. For example, apparent changes in capacity can emerge from learning about specific tasks (MacDonald & Christiansen, 2002). Specific models should prove useful in investigating such potential points of contrast and contrast between approaches.

Another difference between these models concerns the behaviors that were viewed as central for the models to explain. Although capturing stage-like transitions was central to both McClelland’s (1989, 1995) and van Rijn et al.’s (2003) models, these approaches may differ in exactly how stage-like the target behavior was viewed to be. As mentioned, the stage-like progression of McClelland’s model was criticized for not being sufficiently stage-like (Jansen & van der Maas, 1997; Rijmkiares et al., 1996). This criticism was based in part on the statistical technique of latent class analysis, which was used to try to assess the number and kind of rules leading to observed behaviors on balance scale problems. However, this statistical technique and associated conclusions have been criticized on multiple grounds (Siegler & Chen, 2002). The rules identified through the latent class analysis technique can be very unstable, even within a short test session in which problems are presented in a consistent way. Thus, it is not clear that such analyses truly reveal stable rules being used by children. Furthermore, these techniques have been used to make claims about the abruptness of transitions between rules; however, they would need to be applied to longitudinal data to make such claims, which has not been done.

Other differences between the models might be reconciled in a relatively straightforward way. The van Rijn et al. (2003) model is presented as unique in accounting for transitions between rules in the absence of feedback. This represents an important advance, given that other
models had not simulated this effect. However, other models might be able to accommodate such findings in much the same way that van Rijn et al.'s model did. Van Rijn et al.'s model increased activation of the distance concept with increasing distance differences; in the same way, if McClelland’s model were to increase activation of distance processing units, it might lead to similar results. Such a finding would be consistent with the manipulation and conclusions from the van Rijn et al. model. Another possibility is that different models might provide different explanations, or different levels of explanation, for the finding that children can transition between rules in the absence of feedback. For example, neural networks might account for such findings through self-organizing learning algorithms (e.g., Hebbian learning algorithms) that do not rely on feedback.

Language: The Past Tense

In addition to progressing through different stages as described in the balance scale case, children sometimes show U-shaped curves in their developmental trajectories. As these children learn, they first get worse at a task, moving from a higher level of performance to a lower level, and then ultimately progressing back up to a higher level. This kind of behavior should provide important constraints on theories of development, and has been the subject of much discussion (e.g., Zelazo, 2004).

In the case of language learning, children can show a U-shaped learning curve in their production of the past tense inflection for irregular verbs, such as “go.” They may initially produce the correct inflection (“went”), but then go through a period where they make overregularizations (saying “goed”), and finally produce correct inflections again. Why children show such U-shaped curves in the learning of the past tense has been the source of considerable debate (e.g., Marcus et al., 1992; McClelland & Patterson, 2002b; Pinker & Prince, 1988; Pinker & Ullman, 2002; Plunkett & Marchman, 1993). In what follows, we consider answers provided through investigations with neural network (e.g., Rumelhart & McClelland, 1986a) and production system models (Taatgen & Anderson, 2002).

Neural Network Models

Neural network models of past tense learning have been used to investigate how U-shaped learning curves might arise through a single representational system, which handles both regular and irregular verbs (Daugherty & Seidenberg, 1992; Hare & Elman, 1992; MacWhinney & Leinbach, 1991; Plunkett & Juola, 1999; Plunkett & Marchman, 1991, 1993, 1996; Rumelhart & McClelland, 1986a). Such models have been presented with the task of mapping word stems (such as “go”) onto their past tense forms (“went”), and have learned through error-driven algorithms. Thus, they learned from changes to connection weights that reduced the discrepancies between their outputs and target outputs. Although children do not often receive explicit error feedback on their syntactic errors (Pinker, 1984), they might receive implicit feedback by comparing their guesses about what inflected form they would hear with what they actually heard.

Rumelhart and McClelland (1986a) presented the first attempt to explain the U-shaped overregularization curve in a single representational system. This model consisted primarily of two layers that mapped a phonological input representation of a word stem to a phonological output representation of the past tense of that stem. With repeated exposures to words, connections in this network were adjusted until the model was able to produce the past tense forms of both regular verbs and exceptions. The network also included a fixed encoding network on the input side that converted representations from a string of individual phonemes to conjunctive representations of phonemes, and a fixed decoding network on the output side that performed the reverse conversion. In this single system, the same units and connections were used for processing regulars and exceptions. As a result, the model naturally handled a key aspect of exception words: They tend to show similarities with regular words, and thus have been termed “quasi-regular” (McClelland & Patterson, 2002b; Plaut, McClelland, Seidenberg, & Patterson, 1996). The majority of exception past tenses in English end in /ed/ or /t/ (e.g., had, told, cut, slid, taught), as do regular past tenses. Of the exception past tenses that do not show this pattern, most are quasi-regular in other ways, such as in preserving the consonants from the stem in the past tense form (e.g., sing-sang, rise-rose). Given this structure to the language, the model could leverage the same units and connections developed for mapping the regular past tense in forming past tenses of exception words.

The model also simulated the U-shaped overregularization curve (Figure 10.6); however, it was heavily criticized for relying on questionable manipulations in the training set for this effect (Pinker & Prince, 1988). The model was initially trained on a corpus of 10 verbs (8 ir-