HANDBOOK OF CHILD PSYCHOLOGY

FIFTH EDITION 1998

Volume 2: Cognition, Perception, and Language

Editor-in-Chief

WILLIAM DAMON

Volume Editors

DEANNA KUHN and ROBERT S. SIEGLER

John Wiley & Sons, Inc.
New York • Chichester • Weinheim • Brisbane • Singapore • Toronto

CHAPTER 13

Information Processing

DAVID KLAHR and BRIAN MACWHINNEY

SEARCHING FOR MECHANISMS OF TRANSITION AND CHANGE

In every field of science, questions about transition and change have challenged generations of researchers. In physics, the goal is to understand the processes involved in the origin of the universe. In biology, researchers attempt to discover the processes underlying cell differentiation, growth, and death. In the field of cognitive development, the fundamental questions are about the structure and content of children's knowledge, and the nature of the transition mechanisms that allow the child to move between progressive knowledge states.

Developmentalists have long sought an adequate language for formulating these questions and for proposing answers to them. Vygotsky (1962) viewed inner speech as a way of building temporary mental representations, but he never specified how these temporary representations could lead to long-term developmental changes. Piaget sought to explain knowledge structures and transition processes by adapting the formalisms available at the time. Using logic and mathematics, he constructed a representational system (Piaget, 1953) and from biology he borrowed the notion of assimilation and accommodation (Piaget, 1975). However, subsequent researchers have found these constructs to be exasperatingly ambiguous (Brainerd, 1978; Cohen, 1983; Klahr, 1982; Miller, 1983).

About 30 years ago, with the emergence of what came to be known as "the information processing approach" (see McCorduck, 1979; Palmer & Kimchi, 1986 for a succinct history), a new set of conceptualizations and methodologies were proposed as a means of addressing questions about cognitive development. In the past three decades, most of what has been discovered about children's thinking deals, in one way or another, with how they process information. Today, few psychologists would disagree with the

Preparation of this chapter was supported in part by a grant from the National Institute of Child Health and Human Development (R01-HD25211) to the first author and by a grant from the National Institute for Deafness and Communicative Disorders (R01-DC01903) to the second author.

claim that cognitive development involves changes in the content, structure, and processing of information.

However, beyond this diffuse consensus, there is substantial diversity in the kinds of answers that different information-processing researchers would give to more focused questions, such as: What do we mean by information and by processing? What is the stuff that gets processed? What are the characteristics of the processor? With what information and what processes is the neonate endowed? Which of these change with development? In this chapter, we focus on the way in which one particular subgroup of researchers—sometimes characterized as members of the "hard core" information-processing camp (see Klahr, 1992 for an overview)—has used computational models to suggest answers to such questions. Thus, from the very broad topic of information processing, we limit our discussion to developmentally relevant computational models of cognition and language.

First, we provide a short historical account of the emergence of computational approaches to studying cognitive development. Then we discuss three broad classes of computational models and provide a brief overview of each. Following that, we describe in depth the two most widely used types of computational models: production systems and connectionist systems. Finally, we close with a comparison of the two approaches and with some speculations about the future of computational modeling.

PRECURSORS OF COMPUTATIONAL MODELS OF DEVELOPMENT

More than 30 years ago, Herbert Simon—one of the founders of the cognitive revolution, but not a cognitive developmentalist—sketched the path that a computational approach to cognitive development might take:

> 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)

Simon's suggestion contained two ideas that departed radically from the then-prevailing views in developmental psychology. The first idea was that cognitive theories could be stated as computer programs. These "computational models of thought," as they have come to be known, have one important property that distinguishes them from all other types of theoretical statements: They independently execute the mental processes they represent. That is, rather than leaving it to the reader to interpret a verbal description of such processes as encoding an external stimulus or searching a problem space, computational models actually do the encoding or searching. Consequently, the complex implications of multiple processes can be unambiguously derived.

The second idea in Simon's suggestion followed from the first: If different states of cognitive development could be described as programs, then the developmental process itself could also be described as a program that took the earlier program and transformed it into the later one. Such a program would have the capacity to alter and extend its own processes and structures. That is, it would be a computational model possessing some of the same self-modification capacities as the child's developing mind.

This two-step view—proposing a performance model and then seeking an independent set of "transition mechanisms" that operate on that performance model—was influential in the early years of computational modeling of cognitive development (Baylor & Gascon, 1974; Klahr & Wallace, 1976; Young, 1976) and we have included several examples of it in this chapter. However, over the years, the sharp distinction between performance models and learning models has become blurred. Today the most promising approaches are those that formulate computational models that are always undergoing self-modification, even as they perform at a given "level" or "stage."

Classes of Computational Models

It is not easy to construct computational models that achieve an appropriate balance between performance and adaptation. Consequently, two relatively distinct—and at times adversarial—approaches to computational modeling of developmental phenomena have emerged. We will describe these two broad classes of systems: production systems and connectionist systems.[1] They approach issues of performance and adaptation from different points of departure. In

[1] This dichotomy has also been characterized as between symbolic models (production systems) and subsymbolic (connectionist

general, production systems emphasize performance over adaptation, while connectionist systems emphasize adaptation over performance. However, as we will explain later in the chapter, the distinctions between the two approaches are diminishing as both fields devote more effort to addressing developmental issues. Most of the rest of this chapter will be aimed at clarifying and defending this assertion. In addition, we will briefly describe some computational efforts that are neither production systems nor connectionist systems.

Production Systems

One of the paradoxes of cognition is that it is simultaneously serial and parallel. Massive amounts of parallelism are manifest both deep within the system at the neural level as well as at the surface where the organism's perceptual and motor systems interact with the environment. Paradoxically, rational thought, attention, and motor acts, from speech to locomotion, require a nontrivial degree of seriality. For example, if your phone rang while you were reading this chapter, you would immediately consider what to do about it: Pick up the receiver? Let your answering machine screen the call? Ask someone else to answer the phone? Your mind must contain some rules that can respond to this kind of unexpected input, while at the same time containing other rules that enable you to systematically and sequentially scan the page from left to right and top to bottom (with necessary regressions) while reading. What kind of processing system can account for these phenomena? Production-systems models of cognition were invented as a response to this challenge (Newell & Simon, 1972).

Production systems are a class of computational models consisting of two interacting data structures:

1. A *working memory* consisting of a collection of symbol structures called working memory *elements*.
2. A *production memory* consisting of condition-action rules called *productions*, whose conditions describe configurations of working memory elements and whose actions specify modifications to the contents of working memory.

Production memory and working memory are related through the *recognize-act* cycle, which consists of three distinct processes:

1. The *recognition* (or matching) process finds productions whose conditions match against the current state of working memory. Because the components of a production's conditions are usually stated as variables, a given production may match against working memory in different ways, and each such mapping is called an instantiation. Moreover, several different productions may be instantiated (or satisfied) at once.
2. The *conflict resolution* process determines which instantiated productions will be applied (or fired).
3. The *act* process applies the instantiated actions of the selected productions. Actions can include the modification of the contents of working memory, as well as external perceptual-motor acts.

The recognize-act process operates iteratively. As productions fire, the contents of working memory change. This leads to another recognition cycle, which leads to a different set of productions being satisfied.[2]

Production systems can be thought of as collections of complex, dynamic systems of stimulus-response (S-R) pairs. The S corresponds to the condition sides of the productions that search, in parallel, structures in working memory. The relation between the working memory of production systems and the working memory construct in experimental psychology has always been somewhat vague. Production-system working memory has been variously conceptualized as short-term memory (Waugh & Norman, 1965), M-space (Pascual-Leone, 1970), short-term plus intermediate-term memory (Bower, 1975; Hunt, 1971), the currently activated portion of long-term memory, or simply as the current state of awareness of the system. More recent models of working memory (Baddeley, 1986, 1990) are more complex than the initial "box of slots" conceptualizations. However, regardless of the mapping between these theoretical constructs and the working memory of production-system architectures, the effect of this architectural feature is clear. The immediacy and recency of information that can satisfy productions serves to maintain context, while still admitting of abrupt shifts in attention, if either internal processing or perceptual input effects relevant changes in that context. Thus, production systems

systems). The subsymbolic characterization was introduced by connectionists who view their models as explanations of the "micro-structure of cognition" (Rumelhart & McClelland, 1986).

[2] Many of the constraints implicit in this simple description have been relaxed and modified during the 20 years or so that production-system architectures have been under development. We will describe these developments in the main section on production systems.

resolve the parallel-serial paradox by providing a parallel associative recognition memory on the condition side and a serial response on the action side.

Connectionist Systems

Connectionist models share a set of assumptions about the nature of neural computation: its connectivity, its representation of knowledge, and the rules that govern learning. Connectionist systems use neither symbols nor rules to manipulate those symbols. The basic premises in these systems are inspired by our knowledge of how the brain is "wired." Connectionist systems consist of elementary nodes or units, each of which has some degree of activation. Nodes are connected to each other in such a way that active units can either excite or inhibit other units. Connectionist networks are dynamic systems that propagate activation among units until a stable state is reached. Information or knowledge is represented in the system not by any particular unit, but rather by the pattern of activation over a large set of units, any one of which may participate to some degree in representing any particular piece of knowledge. McClelland (1995, p. 158) succinctly characterizes the essence of these models:

> On this approach—also sometimes called the parallel-distributed processing or PDP approach—information processing takes place through the interactions of large numbers of simple, neuron-like processing units, arranged into modules. An active representation—such as the representation one may have of a current perceptual situation, for example, or of an appropriate overt response—is a distributed pattern of activation, over several modules, representing different aspects of the event or experience, perhaps at many levels of description. Processing in such systems occurs through the propagation of activation among the units, through weighted excitatory and inhibitory connections.
>
> As already suggested, the knowledge in a connectionist system is stored in the connection weights: it is they that determine what representations we form when we perceive the world and what responses these representations will lead us to execute. Such knowledge has several essential characteristics: First, it is inchoate, implicit, completely opaque to verbal description. Second, even in its implicit form it is not necessarily accessible to all tasks; rather it can be used only when the units it connects are actively involved in performing the task. Third, it can approximate symbolic knowledge arbitrarily closely, but it may not; it admits of states that are cumbersome at best to describe by rules; and fourth, its acquisition can proceed gradually, through a simple, experience-driven process.

Because connectionist systems are inherently learning systems, the two-step approach (first performance models, then transition models) has not been used. Instead, designers of connectionist models have focused on models that learn continuously, and they have attempted to illustrate how different distributions of connectivity among the nodes of their networks correspond to different knowledge levels in children. The earliest applications were in the area of language acquisition, but more recent models have begun to examine conceptual development and problem solving.

Ad Hoc Models

In many cases, a researcher may have a theory about some phenomenon that is sufficiently complex that only a computational model will enable one to derive predictions from it. However, the modeler may not be prepared to make a commitment to the theoretical claims of either connectionist or production-system approaches. In such cases, one simply chooses to focus on the knowledge structures and computational processes, and employs an ad hoc computational architecture in which to formulate and run the model. This approach enables the model builder to focus on the complexities of the domain under consideration without being constrained by global architectures or particular learning algorithms. One advantage of ad hoc systems is that, because they are not constrained by global theoretical concerns, they often achieve extremely precise and fine-grained fits to empirical measures of children's performance. The disadvantages are that their range of application is relatively narrow, and their relation to the total cognitive system is not specified.

Example: A Computational Model for Children's Strategy Choice in Arithmetic. The focus of this chapter is on production systems and connectionist systems. However, in order to demonstrate the way in which computational models can enhance our understanding of developmental phenomena even when there is no strong commitment to a particular cognitive architecture, we will describe one such model.

Siegler and his colleagues have developed a series of computational models to account for children's performance on simple addition problems (Siegler & Shipley, 1995; Siegler & Shrager, 1984). The basic phenomenon is that children use a variety of strategies to solve problems such as $3 + 4$. One strategy is to simply retrieve the answer from memory. Another is to start a count at 4, and then count up 3 steps to 7. Yet another is to count on their fingers: first three fingers, then four fingers, and then to count all the extended fingers. The relation between the use of these strategies and their speed and accuracy is highly systematic. Siegler's models address two basic questions. How do each of the distinct strategies work? How do children choose among them?

The key feature in the first computational model (Siegler & Shrager, 1984) was a data structure in which, for every pair of integers, there was a distribution of associations to possible answers (both correct and incorrect). For example, the problem 3 + 5 has associated with it not only 8, but also other possible responses that might have been given in the past, just as 6, 7, and 9. The distribution of response strengths to possible answers gives each problem a characteristic shape, and these distributions can be classified along a dimension of *peakedness*. In a problem with a peaked distribution, most of the associative strength is concentrated in a single answer, ordinarily the correct answer. At the other extreme, in a *flat distribution*, associative strength is dispersed among several answers, with none of them forming a strong peak.

This data structure is then used by the model to decide whether to produce an answer through direct retrieval—with the response determined by the probability distribution associated with that problem, or via some other, more deliberate counting-based strategy. When run on a variety of data sets, the model provided an excellent fit to both speed and accuracy data from children's performance on the same type of problems that were presented to the model. In fact, it facilitated the derivation of some nonintuitive predictions about the correlation between error rates and strategy selection that were supported by the empirical results. Moreover, the model challenges the notion that metacognitive processes play a role in children's choice of addition strategies. Instead, intelligent strategy choices emerge from the application of simpler, more basic processes. This kind of "emergent property" is a particularly important feature of computational models, and we will discuss it further at the end of this introductory section.

But the model had its shortcomings. Siegler and Shipley (1995) were able to analyze its behavior with extreme precision and conclude that "it was too inflexible, too limited in its explicitness, and too dumb." Harsh words, but true. (But could one ever assess a verbally based theory with such exactness?)

In order to remedy this problem, Siegler and Shipley formulated a second computational model—the Adaptive Strategy Choice Model (ASCM). Their goal was to create a more flexible, more precise, and more intelligent model of strategy choice. In ASCM, each strategy has associated with it a database containing information about its accuracy, speed, and novelty, as well as its *projected* accuracy, speed and special features. The improved model was designed to account for variability in strategies, answers, and individual performance patterns, as well as the order in which strategies were considered. ASCM was able to make adaptive choices on novel as well as familiar problems and to make good choices among its alternative strategies.

Thus, ad hoc computational models offer theoretical advances, even though they do not entail the more global systemic assumptions of either production-system or connectionist frameworks. Indeed, in a review of over a score of computational models relevant to cognitive development, Rabinowitz, Grant, and Dingley (1987) indicate the influential role of ad hoc models. A more recent example of ad hoc computational modeling is the work on analogical reasoning described by Gentner and her colleagues (Gentner, Rattermann, Markman, & Kotovsky, 1995). However, because such models are quite diverse in the assumptions that they make, for the remainder of this chapter we will focus only on production systems and connectionist systems.

Psychological Theory and Computer Simulation

Before leaving this preliminary section, we will make a few general comments about computational modeling as a form of theory building in psychology. In particular, we want to address a common misconception about the role of the computer in psychological theory.

Critics of the hard-core information-processing approach often attribute to computational modelers the belief that the digital computer is an appropriate model for the mind. For example, Ann Brown (1982) correctly points out that "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." Although the statement clearly applies to computers, it equally clearly does *not* apply to computational models—even though they are implemented on computers. For example, we have just described how the ASCM model demonstrates adaptive change, and we will describe several other adaptive models later in this chapter.

The misattribution derives from a failure to distinguish between the theoretical content of a program that runs on a computer and the psychological relevance of the computer itself. Hard-core information-processing theories are sufficiently complex that it is necessary to run them on computers in order to explore their implications. But this does not imply that a theory bears any necessary resemblance to the computer on which it runs. Meteorologists who run computer simulations of hurricanes do not believe that the atmosphere works like a computer. Furthermore, the same theory could be implemented on computers having radically different underlying architectures and mechanisms.

Note that this distinction between computational models and computers holds not only for symbolically based simulations but also for connectionist simulations. Even such inherently parallel, highly interconnected systems sit atop computing hardware that is statically organized in a fashion bearing no relation at all to the living neural tissue that the human brain comprises.

The goal of computational approaches to cognitive development is to determine the extent to which the emergence of intelligent behavior can be accounted for by a computational system that is manifested in the physical world. Consequently, because both computers and brains are computational systems, some of the theoretical constructs and insights that have come out of computer science may be relevant for cognitive developmental theory.

One such insight is what Palmer and Kimchi (1986) call the *recursive decomposition* assumption: Any nonprimitive process can be specified more fully at a lower level by decomposing it into a set of subcomponents and specifying the temporal and informational flows among the subcomponents. This is a good example of how abstract ideas from computer science have contributed to computational models of psychological processes: "it is one of the foundation stones of computer science that a relatively small set of elementary processes suffices to produce the full generality of information processing" (Newell & Simon, 1972, p. 29). An important consequence of decomposition is that

> . . . the resulting component operations are not only quantitatively simpler than the initial one, but *qualitatively different* from it . . . Thus we see that higher level information-processing descriptions sometimes contain *emergent properties* that lower level descriptions do not. It is the *organization* of the system specified by the flow relations among the lower level components that gives rise to these properties. (Palmer & Kimchi, 1986, p. 52)

The importance of emergent properties cannot be overemphasized, for it provides a route to explaining how intelligence—be it in humans or machines—can be exhibited by systems comprised of unintelligent underlying components—be they synapses or silicon. Even if one defines "underlying components" at a much higher level—such as production systems or networks of activated nodes, emergent properties still emerge, for that is the nature of complex systems.

The emergent property notion provides the key to our belief that computational approaches provide a general framework, particular concepts, and formal languages that make possible the formulation of powerful theories of cognitive development. The fundamental challenge is to account for the emergence of intelligence. Intelligence must develop from the innate kernel. The intelligence in the kernel, and in its self-modification processes, will be an emergent property of the *organization* of elementary (unintelligent) mechanisms for performance, learning, and development.

In the rest of this chapter, we will first describe production systems, and their use in cognitive developmental theory. Next we will give a similar treatment to connectionist systems. Finally, we will discuss the similarities and differences between these two approaches to computational modeling, with special emphasis on issues of direct relevance to cognitive development.

PRODUCTION-SYSTEM ARCHITECTURES

"What is happening in the human head to produce human cognition?" asks John Anderson in the opening chapter of his most recent book on the ACT-R theory of human thought (Anderson, 1993). The question is as fundamental to developmentalists as it is to those who focus on adult cognition. Anderson's answer is unequivocal:

> Cognitive skills are realized by production rules. This is one of the most astounding and important discoveries in psychology and may provide a base around which to come to a general understanding of human cognition (p. 1)

Anderson goes on to point out that

> production systems are particularly grand theories of human cognition because they are cognitive architectures . . . relatively complete proposals about the structure of human cognition. . . . Just as an architect tries to provide a complete specification of a house (for a builder), so a . . . cognitive architecture tries to provide a complete specification of a system. There is a certain abstractness in the architect's specification, however, which leaves the concrete realization to the builder. So too, there is an abstraction in a cognitive or

> computer architecture. One does not specify the exact neurons in a cognitive architecture, and one does not specify the exact computing elements in a computer architecture.[3] (pp. 3–4)

In this section we describe production systems and their relevance and potential for advancing our understanding of cognitive development. Before we get to production systems, as such, a bit of preliminary work is necessary. First we discuss a few issues surrounding the notion of "symbol systems." Then we describe a cognitive architecture that represents the standard view of adult cognition that gained widespread acceptance in the 1970s and 1980s. With those preliminaries out of the way, we describe production systems proper.

Symbol Systems

Production systems represent the most elaborated and extensive examples of what has come to be called (more by its critics than its advocates) the "symbolic approach" to computational modeling of cognition and cognitive development. It is important to provide a brief introduction to the theoretical assumptions inherent in this approach. The role of symbols, symbol structures, and symbol manipulation in computational models is best described by Newell (1980). He defines a physical symbol system as one that

> Is capable of having and manipulating symbols, yet is also realizable within our physical universe . . . [This concept] has emerged from our growing experience and analysis of the computer and how to program it to perform intellectual and perceptual tasks. The notion of symbol that it defines is internal to this concept of a system. Thus, it is a hypothesis that these symbols are in fact the same symbols that we humans have and use everyday of our lives. Stated another way, the hypothesis is that humans are instances of physical symbol systems, and by virtue of this, mind enters into the physical universe. (p. 136)

Perhaps the most remarkable aspect of Newell's characterization of physical symbol systems is that it solves the venerable "mind-body problem." The essential property of a symbol (physically represented in silicon or neurons) is that it can designate something else (represented as a symbol structure). Such symbols comprise the elementary units in *any* representation of knowledge including sensory-motor knowledge or linguistic structures. Moreover, because these representations encode information about the external physical and social world, they have a semantics as well as a syntax.

Philosophical distinctions between dense and articulated symbols (Goodman, 1968) or personal and consensual symbols (Kolers & Smythe, 1984) emphasize the likelihood of idiosyncratic symbol structures for specific individuals, and the difference between internal symbol structures and their external referents. However, they are entirely consistent with Newell's physical symbol system hypothesis.

A First-Order Cognitive Architecture

Most of the work on both symbolic and connectionist cognitive architectures has focused on adult cognition, rather than on cognitive development. Nevertheless, developmentalists interested in a variety of cognitive processes have adopted—either implicitly or explicitly—the general view of the adult information-processing system that emerged in the late 1960s and early 1970s (Atkinson & Shiffrin, 1968; Craik & Lockhart, 1972; Norman, Rumelhart, & Group, 1975). In this section, we describe a system intended to depict the essential cognitive architecture of a normal adult. This standard description includes three major architectural components:

1. Several buffers in which information from various sensory modalities remains briefly active and available for further processing—such as a visual "iconic" memory and an "acoustic buffer";
2. A limited-capacity memory (of from two to seven "chunks" of information) that can retain material for a few seconds if unrehearsed but for much longer if continually rehearsed. As noted earlier, this memory has been variously conceptualized as short-term memory, working memory, and immediate memory. In some models, these distinctions are specific and theoretically important, while in others they are indistinguishable.
3. An (effectively) unlimited, content-addressable long-term memory.

Although this characterization was inspired by, and is analogous to, the gross functional features of computer

[3] Anderson continues: "This abstractness even holds for connectionist models that claim to be 'neurally inspired.' Their elements are in no way to be confused with real neurons. . . ." In the final section of this chapter we will discuss this claim and others related to the neural realism of connectionist models of cognition.

architectures, it also represents an attempt to account for the plethora of empirical findings that have emerged from experimental studies of human information processing. As we have already explained, the analogy to computer architectures in no way rests on the assumption that, at more microscopic levels of underlying hardware, computers bear any resemblance to neural circuitry.

The notion of a cognitive architecture was originated by Newell (1973, 1981) and has since gone through several successive refinements. One of the most detailed is Card, Moran, and Newell's (1983) model of the human information-processing system that includes not only the gross organization of the different information stores and their connections, but also estimates of processing rates and capacities. This Model Human Processor (MHP) was designed to facilitate predictions about human behavior in a variety of situations involving interactions between humans and computers. It was based on a vast amount of empirical data on human performance in perceptual, auditory, motor, and simple cognitive tasks.

The MHP is illustrated in Figure 13.1 and its principles of operation are listed in Table 13.1. It includes a long-term memory; a working memory; two perceptual stores for visual and auditory information; and three subsystems for cognitive, motor, and perceptual processing. For each of these stores, there are associated estimates of storage capacity, decay times, cycle times, and the type of code as well as connectivity to the rest of the system.

The perceptual system consists of sensors and associated buffer memories, the most important buffer memories being a Visual Image Store and an Auditory Image Store to hold the output of the sensory system while it is being symbolically coded. The cognitive system receives symbolically coded information from the sensory image stores in its Working Memory and uses previously stored information in Long-Term Memory to make decisions about how to respond. The motor system carries out the response.

For some tasks (pressing a key in response to a light), the human must behave as a serial processor. For other tasks (typing, reading, simultaneous translation), integrated, parallel operation of the three subsystems is possible, in the manner of three pipe-lined processors: information flows continuously from input to output with a characteristically short time lag showing that all three processors are working simultaneously. The memories and processors are described by a few parameters. The most important parameters of a memory are: μ, the storage capacity in items; d, the decay time of an item; and k, the main code type (physical, acoustic, visual, semantic). The most important parameter of a processor is t, the cycle time (Card et al., 1983, pp. 24–25).

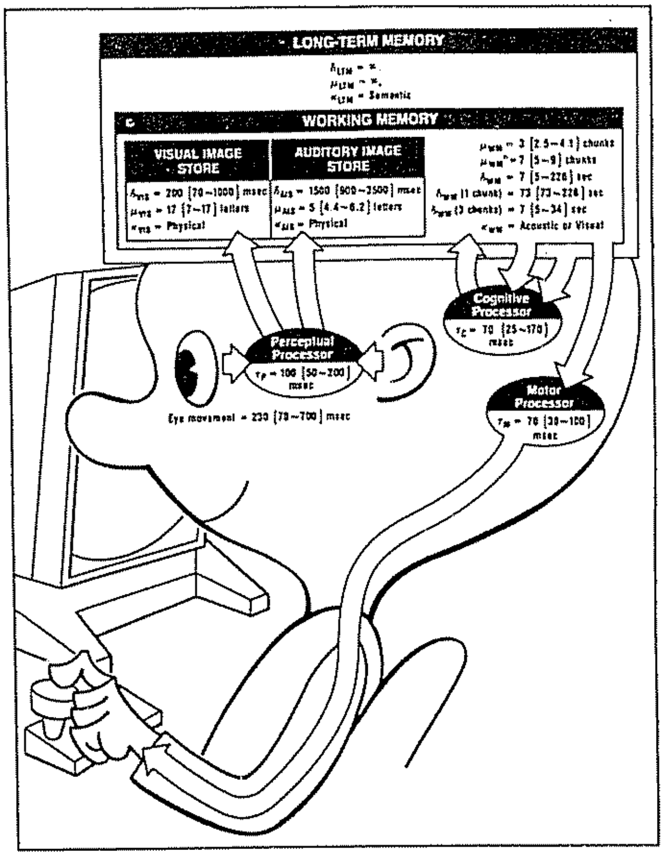
LONG-TERM MEMORY
$\delta_{LTM} = \infty$
$\mu_{LTM} = \infty$
κ_{LTM} = Semantic
WORKING MEMORY
VISUAL IMAGE STORE
δ_{VIS} = 200 [70~1000] msec
μ_{VIS} = 17 [7~17] letters
κ_{VIS} = Physical
AUDITORY IMAGE STORE
δ_{AIS} = 1500 [900~3500] msec
μ_{AIS} = 5 [4.4~6.2] letters
κ_{AIS} = Physical
μ_{WM} = 3 [2.5~4.1] chunks
μ_{WM}^{*} = 7 [5~9] chunks
δ_{WM} = 7 [5~226] sec
δ_{WM} (1 chunk) = 73 [73~226] sec
δ_{WM} (3 chunks) = 7 [5~34] sec
κ_{WM} = Acoustic or Visual
Perceptual Processor
τ_P = 100 [50~200] msec
Cognitive Processor
τ_C = 70 [25~170] msec
Motor Processor
τ_M = 70 [30~100] msec
Eye movement = 230 [70~700] msec

Figure 13.1 The model human processor—memories and processors. Sensory information flows into Working Memory through the Perceptual Processor. Working Memory consists of activated chunks in Long-Term Memory. From *The Psychology of Human-Computer Interaction* by S. Card, T. P. Moran, & A. Newell, 1983. Hillsdale, NJ: Erlbaum. Copyright © 1983 by Lawrence Erlbaum Associates. Reprinted with permission.

The MHP was formulated to account for the perceptual and motor behavior of adults interacting with computers, and it has been used successfully to help design and evaluate human-computer interfaces (Gray, John, & Atwood, 1993). It was definitely *not* designed to advance developmental theory. Nevertheless, we include it here because we believe that it exemplifies the more general attempt to formulate a cognitive architecture that is consistent with the massive amount of empirical data on human performance. As such, it presents an obvious challenge to developmentalists: What would an MHP look like for an infant, for a preschooler, or for an adolescent? Which of the architectural features, processing rates, and information capacities

TABLE 13.1 Principles of Operation for the Model Human Processor (From Card et al., 1983)

P0. *Recognize-Act Cycle of the Cognitive Processor.* On each cycle of the Cognitive Processor, the contents of Working Memory initiate actions associatively linked to them in Long-Term Memory; these actions in turn modify the contents of Working Memory

P1. *Variable Perceptual Processor Rate Principle.* The Perceptual Processor cycle time t_r varies inversely with stimulus intensity

P2. *Encoding Specificity Principle.* Specific encoding operations performed on what is perceived determine what is stored, and what is stored determines what retrieval cues are effective in providing access to what is stored.

P3. *Discrimination Principle.* The difficulty of memory retrieval is determined by the candidates that exist in the memory, relative to the retrieval clues.

P4. *Variable Cognitive Processor Rate Principle.* The Cognitive Processor cycle time t_c is shorter when greater effort is induced by increased task demands or information loads; it also diminishes with practice.

P5. *Fitt's Law.* The time T_{pos} to move the hand to a target of size S which lies a distance D away is given by:

$$T_{pos} = I_M \log_2(D/S + .5),$$

where $I_M = 100\ [70 \sim 120]$ msec/bit.

P6. *Power Law of Practice* The time T_n to perform a task on the nth trial follows a power law:

$$T_n = T_1 n^{-a},$$

where $a = .4\ [.2 \sim .6]$.

P7. *Uncertainty Principle.* Decision time T increases with uncertainty about the judgment or decision to be made:

$$T = I_C H$$

where H is the information-theoretic entropy of the decision and $I_C = 150(0 \sim 157)$ msec/bit. For n equally probable alternatives (called Hick's Law),

$$H = \log_2 (n + 1)$$

For n alternatives with different probabilities, p_i, of occurrence,

$$H = \Sigma_i\, p_i \log_2 (1/p_i + 1)$$

P8. *Rationality Principle.* A person acts so as to attain his goals through rational action, given the structure of the task and his inputs of information and bounded by limitations on his knowledge and processing ability:

Goals + Task + Operators + Inputs + Knowledge
+ Process-limits Æ Behavior

P9. *Problem Space Principle.* The rational activity in which people engage to solve a problem can be described in terms of (1) a set of states of knowledge, (2) operators for changing one state into another, (3) constraints on applying operators, and (4) control knowledge for deciding which operator to apply next.

that so effectively characterize a normal adult would have to change in these "kiddy" versions of the MHP? Finally, what additional features would an MHP need to have in order to develop from the neonate processor to the adult version?

The answers to such questions will occur at two levels. At the level of rates and parameters, the kind of results emerging from Kail's (1988, 1991) extensive chronometric studies may ultimately inform theories about the developmental course of the basic cognitive processes that support the system architecture. At present, however, the developmental range of variation from childhood to adulthood in things such as visual scanning rates, STM scanning rates, and so on, is of the same order of magnitude as the MHP estimates for adults, so the experimental results do not constrain the computational models. For example, Card and colleagues estimate a basic cycle time for the MHP between 25 and 170 msec, while Kail's results show a STM scanning rate that varies from about 125 msec per item for eight-year-olds to 50 msec per item for adults. In order for the chronometric results to constrain the broader architectural theories, it will be necessary to combine such "hardware" estimates with more detailed task analyses of the "software" that utilizes the hardware.

At the architectural level, it is necessary to go beyond the global characterizations provided by the MHP and create computational models that utilize that architecture. The creation of production-system models of children's cognitive processes represents a path toward that goal. In the next section we describe this kind of work.

Production Systems for Different Knowledge States

Production systems were first used in computer science without any particular reference to human cognition.[4] With respect to cognitive modeling, the most important early developments include Newell's first computer implementation of a production-system language called PSG (1973) and Anderson's production systems for adult cognitive tasks (Anderson, 1983; Anderson, Kline, & Beasley, 1978) which combined production systems with semantic nets. These systems took the standard model—described in the previous section—and transformed it into specific proposals about the dynamics of the human cognitive architecture.

At their inception, production systems were used to account for adult problem-solving performance in situations requiring a systematic accumulation of knowledge about a problem space as well as an ability to opportunistically change the focus of the problem (Newell & Simon, 1972).

[4] A brief history of production systems—both in computer science and in psychology—is presented in Neches, Langley, and Klahr (1987).

Although the models cast in this form were able to dynamically accumulate knowledge, revise it, and eventually construct a solution to a problem, these initial models did not themselves change. However, issues of change, learning, adaptation—more generally termed "self-modification"—quickly became an issue, and the early literature on production systems includes work on "adaptive production systems" (Waterman, 1975).

The initial use of production systems by developmentalists was the "sequence of models" approach. As noted earlier, the approach seeks to produce a sequence of production-system models for a specific task such that each model represents a different level of performance. The premise was that once such a sequence of models had been created and validated, it would be possible to examine the differences between successive models in order to infer what a transition mechanism would have to accomplish. Although it is now clear that the clean distinction between performance models and transition models was an oversimplification, such models constituted most of the early work in this area, so we will describe a few of them here.

An Example from Conservation

Consider the moment, repeated tens of thousands of times by developmental investigators, when a child is asked the "crucial" conservation question: "Are there more objects in this row, or in this row, or do the two rows have the same number of objects?" How can we represent the mental computations that the child performs in attempting to reply?

Klahr and Wallace (1976) approached this question by formulating a series of increasingly complex production-system models to account for children's understanding of quantitative concepts, starting with models for encoding discrete quantities via subitizing and counting, and ending with children's ability to understand questions about class inclusion, transitivity, and conservation. Their most "mature" model contains productions dealing with several different levels of knowledge. At the highest level are productions that represent general conservation rules, such as "If you know about an initial quantitative relation, and a transformation, then you know something about the resultant quantitative relation." At the next level are productions representing pragmatic rules, such as "If you want to compare two quantities, and you don't know about any prior comparisons, then quantify each of them." At an even lower level are rules that determine which of several quantification processes will actually be used to encode the external display (e g., subitizing, counting, or estimation). Finally, at the lowest level, are productions for carrying out the quantification process.

Later in this chapter, we will describe the most recent version of a production-system model of conservation knowledge (Simon & Klahr, 1995). In this section, we use the earlier work to illustrate some important aspects of production-system models. Table 13.2 lists a few of the key productions from the Klahr and Wallace (1976) model.

At first glance, and when read in the pseudo-formalism used in Table 13.2, the first three productions appear to be nearly identical. But there are important differences between them. P1 corresponds to a situation in which the system has no goals with respect to quantitative comparison, but has just received some external query (from the ubiquitous conservation investigator). P2 corresponds to a situation in which there *is* a goal of determining a relationship, so it establishes the first subgoal along the path to such a determination, which is to compare the quantity of the two collections. Eventually, the system will determine that relationship, and when it does, P3 will notice that it has both a goal to determine a relationship and the requisite information to satisfy that goal.

Now let us consider in more detail just what each of these productions does. P1 detects an element in working memory that results from the encoding of a verbal query. The linguistic processing is not modeled here, but the

TABLE 13.2 Some Productions for Quantity Conservation. Italicized terms represent variables whose values will be determined by the working memory elements that they happen to match. (Adapted from Klahr & Wallace, 1976, Chapter 5)

P1: If you have been asked about a *quantitative relationship* between *collection X* and *collection Y*
then set a goal to determine the relationship between *collection X* and *collection Y*.

P2: If the goal is to determine a *quantitative relationship* between *collection X* and *collection Y*
then set the goal of comparing *collection X* and *collection Y*.

P3: If the goal is to determine a *quantitative relationship* between *collection X* and *collection Y*
and you know a *relationship* between *collection X* and *collection Y*
then respond by saying the *relationship*.

P4: If your goal is to apply knowledge about quantity conservation
and you know that *collection X* and *collection Y* were quantitatively equivalent
and that *collection Y* underwent a quantity-preserving transformation, changing *Y* into *Y'*
then you know that *collection X* and *collection Y'* are quantitatively equivalent.

assumption is that a variety of questions would produce the three pieces of information: one about the relationship, and the other two about the identity of the focal collections. This production illustrates what is meant by "multiple instantiations" of the same production. This production could be satisfied by several different combinations of matching elements, depending on the precise form of the question (e g., Which is longer the top row or the bottom row? or Which is less, the red ones or the blue ones?) and on whether or not there exists in working memory more than one active element that is a member of the *relationship* concept, or the *collection* concept. For example, the second of the two questions above would form the following matches or bindings: *relationship*–less, *collection X*–red ones, *collection Y*–blue ones. But if another referent to a collection—such as "round things" was still active in working memory, another instantiation of the same production might be: less, red, round.

In addition to multiple instantiations of the same production, it is possible (and common) for several different productions to match the contents of working memory at the same time. For example, because the conditions of P2 are a proper subset of the conditions for P3, whenever P3 is satisfied, so is P2. It is up to the conflict resolution process (described below) to decide how to handle such multiple instantiations of the same production, as well the instantiation of more than one production.

P4 is one of several productions in the original model that represent cross products of the three possible relationships between initial quantities (>, =, <) and three classes of transformations (those that effect increases, decreases, or no change). One of the surprising discoveries that came out of the formulation of production-system models of the conservation task was that there were many different kinds of knowledge required before a child could really be said to "have" quantity conservation. Another was that, in addition to productions about conservation as such, the system needs a large number of quantity-specific problem-solving productions, of the sort represented in Table 13.2, that establish the requisite information and the appropriate goal structure for correctly responding to a conservation query.

Knowledge States for Balance Scale Predictions

Klahr and Siegler (1978) used production systems in a different way: to take a noncomputational information-processing model that had already shown an excellent fit to children's performance and recast it as a production-system in order to get a better idea of its dynamic properties.

The production-system models were based on earlier investigations of children's performance on Piaget's balance scale prediction task. Siegler (1978, 1976) proposed an elegant analysis of rule sequences characterizing how children (from 3 years to 17 years old) make predictions on this task (as well as in several other domains having a similar formal structure). This work has provided the basis for many subsequent empirical and theoretical analyses, including computational theories cast as both production systems and connectionist networks. Because we will be discussing these models in some detail, we next describe the balance scale task on which they are based.

The type of balance scale used consisted of a two-arm balance, with several pegs located at equal intervals along each arm. Small circular disks, all of equal weight, were placed on the pegs in various configurations, while the balance was prevented from tipping. The child's task was to predict the direction in which the balance scale would move if it were allowed to.

The basic physical concept that underlies the operation of the balance scale is torque: The scale will rotate in the direction of the greater of the two torques acting on its arms. The total torque on each arm is determined by summing the individual torques produced by the weights on the pegs, and individual torques are in turn computed by multiplying each weight by its distance from the fulcrum. Since the pegs are at equal intervals from the fulcrum, and the weights are all equal, a simpler calculation is possible. It consists of computing the sum of the products of number of weights on a peg times the ordinal position of the peg from the fulcrum. This is done for each side, and the side with the greater sum of products is the side that will go down. (If they are equal, the scale will balance.)

Siegler (1976) demonstrated that the different levels of knowledge that children have about this task could be represented in the form of a sequence of four increasingly "mature" binary decision trees, depicted in Figure 13.2. A child using Model I considers only the number of weights on each side: If they are the same, the child predicts balance, otherwise he predicts that the side with the greater weight will go down. For a Model II child, a difference in weight still dominates, but if weight is equal, then a difference in distance is sought. If it exists, the greater distance determines which side will go down, otherwise the prediction is balance. A child using Model III tests both weight and distance in all cases. If both are equal, the child predicts balance; if only one is equal, then the other one determines the outcome; if they are both unequal, but on the

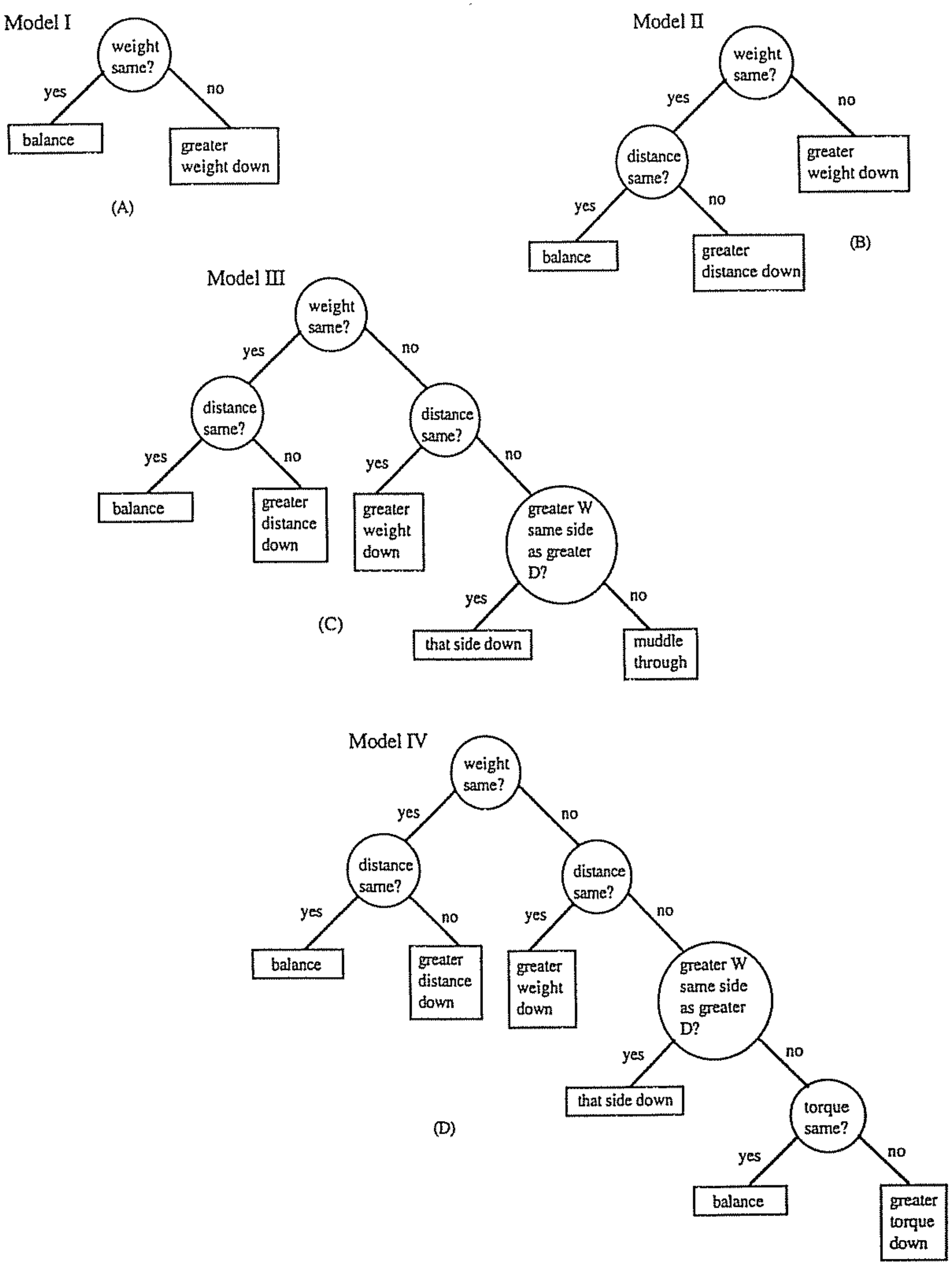

Figure 13.2 Decision tree representation for models I–IV of balance scale predictions. From "The representation of children's knowledge," by D. Klahr & R. S. Siegler, 1978, in H. W. Reese & L. P. Lipsitt (Eds.), *Advances in Child Development and Behavior* (Vol. 12, pp. 61–116). New York: Academic Press. Copyright © 1978 by Academic Press. Reprinted with permission.

same side with respect to their inequality, then that side is predicted to go down. However, in a situation in which one side has the greater weight, while the other has greater distance, a Model III child, although recognizing the conflict, does not have a consistent way to resolve it. This child simply "muddles through" by making a random prediction. Model IV represents "mature" knowledge of the task: Since it includes the sum-of-products calculation, children using it will always make the correct prediction, but if they can base their prediction on simpler tests, they will do so. The components of this knowledge are acquired over a remarkably long span of experience and education. Although children as young as 5 years old usually know that balances such as teeter-totters tend to fall toward the side with more weight, most college students are unable to consistently solve balance scale problems.

Production Systems for Balance Scale Rules. The binary decision trees make clear predictions about the responses that would be made by a child using one of these rules for any specific configuration of weights. However, they are silent on the dynamics of the decision process, and they do not make a clear distinction between encoding processes and decision processes. By recasting the rules as production systems, Klahr and Siegler were able to make a more precise characterization of what develops than was afforded by the decision-tree representation.

The production system is listed in Table 13.3. Consider, for example, Model II in Table 13.3. It is a production system consisting of three productions. The condition elements in this system are all tests for sameness or difference in weight or distance. The actions all refer to behavioral responses. None of the models in Table 13.3 contain a representation for any more detailed knowledge, such as the actual amount of weight or distance, or the means used to encode that information. Nor is there any explicit representation of how the system produces the final verbal output. It is simply assumed that the system has processes—or "operators" that produce encoded representations of the relational information stated in the conditions.

TABLE 13.3 Production System (P) Representations for Models I-IV. D = distance; W = weight. See text for further explanation. (From Klahr & Siegler, 1978)

Model I		
	P1: ((Same W) → (Say "balance"))	
	P2: ((Side X more W) → (Say "X down"))	
Model II		
	P1: ((Same W) → (Say "balance"))	
	P2: ((Side X more W) → (Say "X down"))	
	P3: ((Same W) (Side X more D) → (Say "X down"))	
Model III		
	P1: ((Same W) → (Say "balance"))	
	P2: ((Side X more W) → (Say "X down"))	
	P3: ((Same W) (Side X more D) → (Say "X down"))	
	P4: ((Side X more W) (Side X less D) → muddle through)	
	P5: ((Side X more W) (Side X more D) → (Say "X down"))	
Model IV		
	P1: ((Same W) → (Say "balance"))	
	P2: ((Side X more W) → (Say "X down"))	
	P3: ((Same W) (Side X more D) → (Say "X down"))	
	P4: ((Side X more W) (Side X less D) → (get Torque))	
	P5: ((Side X more W) (Side X more D) → (Say "X down"))	
	P6: ((Same Torque) → (Say "balance"))	
	P7: ((Side X more Torque) → (Say "X down"))	
	Transitional requirements	
	Productions	Operators
I → II	add P3	add distance encoding and comparison
II → III	add P4, P5	
III → IV	modify P4; add P6, P7	add torque computation and comparison

On any recognize-act cycle, only one of these productions will fire, depending on the type of knowledge that the encoding processes have placed in working memory. If the weights are unequal, then P2 will fire; if the weights are equal and the distances are not, then both P1 and P3 will be satisfied, and this "conflict" has to be resolved by the production system architecture. For the production system that Klahr and Siegler proposed, the conflict is resolved by a specificity principle that always selects the more specific of two productions when one is a special case of the other.[5] Finally, if both weights and distances are equal, then only P1 will be satisfied and it will fire. (Note that a production maintains its label across the four models.)

We can compare the four models to determine the task facing a transition model. At the level of productions, the requisite modifications are straightforward: transition from Model I to Model II requires the addition of P3; from Model II to III, the addition of P4 and P5; and from Model III to IV, the addition of P6 and P7 and the modification of P4 to P4′.

Thus far we have compared the models at the level of productions. But productions need information provided by the operators that encode the external configuration.

[5] More recent production-system architectures have dropped the "specificity principle." Conflict resolution will be discussed later in the chapter.

Consequently, it is informative to compare the four models at a finer level of analysis by looking at the implicit requirements for encoding and comparing the important qualities in the environment. The production system for Model I tests for sameness or difference in weight. Thus, it requires an encoding process that either directly encodes relative weight or encodes an absolute amount of each and then inputs those representations into a comparison process. Whatever the form of the comparison process, it must be able to produce not only a same-or-different symbol, but if there is a difference, it must be able to keep track of which side is greater. The production system for Model II requires the additional capacity to make these decisions about distance as well as weight. This might constitute a completely separate encoding and comparison system for distance representations, or it might be the same system except for the interface with the environment.

Model III's production system needs no additional operators at this level. Thus, it differs from Model II only in the way it utilizes information that is already accessible to Model II. The Model IV production system requires a much more powerful set of quantitative operators than any of the preceding models. In order to determine relative torque, it must first determine the absolute torque on each side of the scale, and this in turn requires exact numerical representation of weight and distance. In addition, the torque computation would require access to the necessary arithmetic production systems to actually do the sum of products calculations.

Although we have compared the four models at two distinct levels—productions and operators—the levels are not really that easily separated. Missing from these models is a set of productions that would indicate the interdependence: productions that explicitly determine which encoding the system will make. That is, in these models, there are almost no productions of the form: (want to compare weights) → (attend to stimulus and notice weight). The sole exception to this occurs in P4′ in Model IV. When this model is confronted with a nonconflict problem, either P1, P2, P3, or P5 will fire on the first recognize cycle. For a conflict problem, P4′ fires, and the system attempts to "get torques." The result of this unmodeled action, as described above, would be to produce a knowledge element that could satisfy either P6 or P7 on the next cycle.

Representing the Immediate Task Context. One advantage of a production-system formulation is that it facilitates the extension of a basic model of the *logical* properties of a task to include the processing of verbal instructions, encoding of the stimulus, keeping track of where the child is in the overall task, and so on. For example, in their analysis of *individual* subject protocols on the balance scale, Klahr and Siegler proposed several distinct models to account for some children's idiosyncratic—but consistent—response patterns. Some of these models included not only the basic productions for a variant of one of Siegler's four models for balance scale predictions, but also knowledge about the instantaneous task context.

These models are too detailed to present here. However, it is instructive to consider the way in which such detailed models are able to characterize how much more than balance scale knowledge, as such, is required by a child performing this task. For example, one of the Klahr and Siegler models for an individual subject dealt with the way in which the child maintained, in working memory, the following pieces of information: which side has *more* weight or distance, which side has a *big* weight or distance, what the current criterion value is, what the scale is expected to do, what the scale actually did, whether the prediction is yet to be made or has been made, and whether it is correct or incorrect.

Thus, their model makes a strong claim about how much of the encoded knowledge (i.e., the contents of working memory) must be available at any one moment. Although production-system models do not generally impose any clear constraints on the size of working memory they provide the potential for such an analysis. One of the relatively unexplored areas for future computational modelers is to attempt to integrate the theoretical constructs and empirical results described by working memory capacity theorists, such as Case (1986) and Bidell and Fischer (1994) with the added formalisms and precision of production-system models. Promising steps in this direction are represented by recent work by Halford and his colleagues (Halford, 1993; Halford et al., 1995).

The two examples in this section—conservation and balance scale knowledge—represent the nontransition phase of production system modeling. The primary goal was to explore the nature of the system that could display the different levels of performance observed in the children's responses to these tasks. Thus they exemplify the two-step approach that characterized such early models, even though they did not address the transition process itself. The next step in the progression came in the form of self-modifying production systems.

Production-Systems Approaches to Self-Modification

Many general principles for change have been proposed in the developmental literature. These include equilibration, encoding, efficiency, redundancy elimination, search reduction, self-regulation, consistency detection, and representational redescription. However, such principles are not computational mechanisms. That is, they do not include a specification of how information is encoded, stored, accessed, and modified. It is one thing to assert that the cognitive system seeks equilibration or that a representation is redescribed; it is quite another to formulate a computational model that actually does so.

Adoption of a production-system architecture allows one to pose focused questions about how broad principles might be implemented as specific mechanisms. One way to do this is to assume the role of a designer of a self-modifying production system, and consider the issues that must be resolved in order to produce a theory of self-modification based on the production-system architecture. The two primary questions are:

1. *What* are the basic change mechanisms that lead to new productions? Examples are generalization, discrimination, composition, proceduralization, strengthening, and chunking.
2. *When* are these change mechanisms evoked: when an error is noted, when a production fires, when a goal is achieved, or when a pattern is detected?

Possible Loci of Development in Production-System Architectures

There are two primary classes of changes that can affect the behavior of a production system and each provides a potential site for a partial account of cognitive development. One class of changes is at the level of productions, and involves creating new productions or modifying existing ones. The other class of changes involves the rules of execution of the production system itself. These include changes in the conflict resolution rules and changes in the size or complexity of working memory elements.

Production Changes. One way to generate new productions is to modify the conditions of existing productions. Anderson, Kline, and Beasley (1978) were the first to create production-system models that learned via *generalization* and *discrimination*. The first mechanism creates a new production that is *more* general than an existing production, while retaining the same actions. The second mechanism—discrimination—creates a new production that is *less* general than an existing production, while still retaining the same actions. The two mechanisms lead to opposite results, though in most models they are not inverses in terms of the conditions under which they are evoked.

Various change mechanisms have been proposed that lead to productions with new conditions *and* actions. *Composition* was originally proposed by Lewis (1978) to account for speedup as the result of practice. This method combines two or more productions into a new production with the conditions and actions of the component productions. But conditions that are guaranteed to be met by one of the actions are not included. For instance, composition of the two productions AB→CD and DE→F would produce the production ABE→CDF. The most advanced form of this type of self-modification—chunking—is embodied in the Soar model to be described in the next section.

Another mechanism for creating new productions is *proceduralization* (Anderson, Greeno, Kline, & Neves, 1981). This involves constructing a highly specific version of some general production, based on some instantiation of the production that has been applied. This method can be viewed as a form of discrimination learning because it generates more specific variants of an existing production.

Production System Architecture Changes. As noted earlier, it is often the case that more than a single production is satisfied during the recognition phase of the recognize-act cycle. Thus, conflict resolution offers another decision point at which the behavior of the system can be affected. Production system designers have employed a number of schemes for performing conflict resolution, ranging from simple fixed orderings on the productions, to various forms of weights or strengths (usually based on feedback about the effectiveness of prior production firings), to complex schemes that are not uniform across the entire set of productions, to no resolution at all. Some important aspects of cognitive development, such as attentional increases and the ability to suppress prepotent responses, might be accounted for by developmental changes in these conflict resolution processes.

Another type of architectural change that might be used to explain some aspects of developmental change would be changes in the size and complexity of the working memory elements that can be matched against productions. At present, there are no detailed proposals along these lines, but such an account might provide an integration between existing capacity theories of cognitive development, such as Case (1985), Halford et al. (1995), and computational models of the type described in this chapter.

Chunking and Its Use in a Model of Conservation Acquisition

A basic mechanism for change via chunking was initially proposed by Rosenbloom and Newell (1982, 1987) and first used to explain the power law of practice (the time to perform a task decreases as a power function of the number of times the task has been performed). The learning curves produced by their model are quite similar to those observed in a broad range of learning tasks. The chunking mechanism and the production-system architecture to support it has evolved into a major theoretical statement about the nature of the human cognitive system. The system (called Soar) is one of the most fully-elaborated examples of a complete cognitive theory—a "unified theory of cognition" as Newell (1990) calls it. It would require a substantial extension of the present chapter to give a comprehensive overview of Soar. However, because the Soar architecture has been used in a recently developed theory of conservation acquisition to be described below, we will briefly summarize its main features here.

The Soar architecture is based on formulating all goal-oriented behavior as search in problem spaces. A problem space consists of a set of states and a set of operators that move between states. A goal is formulated as the task of reaching one of a desired set of states from a specified initial state. Under conditions of perfect knowledge, satisfying a goal involves starting at the initial state, and applying a sequence of operators that result in a desired state being generated. Knowledge is represented as productions. When knowledge is not perfect, the system may not know how to proceed. For example, it may not know which of a set of operators should be applied to the current state. When such an impasse occurs, Soar automatically generates a subgoal to resolve the impasse. These subgoals are themselves processed in additional problem spaces, possibly leading to further impasses. The overall structure is one of a hierarchy of goals, with an associated hierarchy of problem spaces. When a goal is terminated, the problem solving that occurred within the goal is summarized in new productions called chunks. If a situation similar to the one that created the chunk ever occurs again, the chunk fires to prevent any impasse, leading to more efficient problem solving.

Soar contains one assumption that is both parsimonious and radical. It is that all change is produced by a single mechanism: chunking. The chunking mechanism forms productions out of the elements that led to the most recent goal achievement. What was at first a search through a hierarchy of subgoals becomes, after chunking, a single production that eliminates any future search under the same conditions. Chunking is built into the Soar architecture as an integral part of the production cycle. It is in continual operation during performance—there is no place at which the performance productions are suspended so that a set of chunking productions can fire. Chunking occurs at all levels of subgoaling, and in all problem spaces. Chunking reduces processing by extending the knowledge base of the system.

Simon and Klahr (1995) used Soar as the theoretical context in which to formulate a computation model of how children acquire number conservation. Their model, called Q-Soar, simulates a training study (Gelman, 1982) in which 3- and 4-year-old children were given a brief training session that was sufficient to move them from the classical nonconserving behavior to the ability to conserve small and large numbers. Q-Soar is designed to satisfy several desirable features of computational models of cognitive development:

1. It is based on a principled cognitive architecture (in this case Newell's Soar theory of cognition).
2. It is constrained by general regularities in the large empirical literature on number conservation.
3. It generates the same behavior as do the children in the specific training study being modeled. That is, it starts out by being unable to pass number conservation tasks, and then, based on the chunks that it forms during the training study, it is able to pass post tests that include both small and large number conservation tests.

Q-Soar asserts that young children acquire number conservation knowledge by measurement and comparison of values to determine the effects of transformation on small collections of discrete objects. Having been shown a transformation on a set of objects, the model first categorizes the transformation. This processing creates new knowledge about this kind of transformation, which becomes available on future occurrences in similar contexts. Eventually, the transformation's effects can be stated without the need for any empirical processing.

Processing Capacity in a Production-System Model

Halford et al. (1995) describe a model of strategy development in transitive inference tasks called the Transitive

Inference Mapping Model (TRIMM). The model—written as a self-modifying production system—is similar to Siegler and Shipley's (1995) ASCM model for strategy choice in arithmetic, in the way it chooses strategies on the basis of their strength (here represented as the strength of productions). In addition, where no strategy is available, TRIMM develops a new strategy by making analogical mappings from earlier representations of situations similar to the current context. One of the novel features of TRIMM is that these mappings are subject to a processing load factor that operates only when new strategies are being developed, but not when existing strategies are adequate. Thus, the model implements and combines both associative and metacognitive mechanisms for strategy development. Once new productions have been formed, they are strengthened or weakened according to their success on the transitive inference tasks presented to the system.

Halford and colleagues make an important observation about the implications of this kind of model for the learning-maturation dichotomy that is so pervasive in discussions about cognitive development.

> It is obvious enough that the question of cognitive development cannot be a matter of learning *or* maturation. However, it is equally inappropriate to propose the question in any other form. For example, it makes no sense to ask whether cognitive development is a matter of capacity *or* knowledge acquisition, capacity *or* expertise, capacity *or* relational encoding, and so on. All of these are really alternate forms of the learning or maturation question. We take it as self-evident that experience-driven processes such as accumulation and organization of a knowledge base, skill acquisition, and efficient encoding, are all important in cognitive development. Modeling some of those processes in detail is what . . . [computational modeling] is about. The question of capacity is not whether it is an alternative to any of these processes, but whether, and how, it interacts with them. (p. 124, emphasis added)

Necessary and Sufficient Mechanisms

Thus far, we have described two classes of potential changes in production systems that can be used to account for developmental phenomena: changes at the level of productions and changes at the level of the production-system architecture. We have illustrated a handful of examples of production systems that use such processes on familiar tasks from the cognitive development literature. But production system modelers have a much more ambitious goal: to explain cognitive development "in the large," rather than on a task by task basis. Indeed, this is one of the reasons why more recent work tends to use production-system architectures that derive from overarching cognitive theories such as Newell's SOAR or Anderson's ACT-R.

One of the fundamental research questions in this area is the extent to which the self-modification processes included in such theories are necessary and/or sufficient to explain cognitive development. For example, it is not yet clear whether the "basic" production modification processes described earlier—such as generalization, discrimination, composition, proceduralization, and chunking—can account for the apparent reorganization necessary to get from novice to expert level (Hunter, 1968; Larkin, 1981; Lewis, 1981; Simon & Simon, 1978). Such reorganization may involve much more than refinements in the productions governing *when* sub-operations are performed. These refinements could be produced by generalization and discrimination mechanisms. However, producing a new procedure requires the introduction of new operators that, in turn, may require the introduction of novel elements or goals—something that generalization, discrimination, composition, and chunking are not clearly able to do.

Some additional mechanisms and processes have been proposed, but they remain to be implemented in computational models. For example, Wallace, Klahr, and Bluff (1987) proposed a novel production-system architecture that included a hierarchically-organized set of nodes, each of which is a semi-autonomous production system, communicating via a shared working memory. Each of these nodes can be simultaneously activated. The basic developmental process involved the construction of new nodes by processing a representation of episodic sequences for the systems' previous behavior (the time line). Another example of a plausible concept that remains to be computationally implemented is Karmiloff-Smith's (1992) "representational redescription"—a process in which the underlying engine of cognitive development involves increasingly efficient reorganizations of knowledge structures and the processes that operate upon them.

Such "soft-core" notions presents challenges to the "hard-core" approach: either implement these ideas, or show that they are theoretically unnecessary, or create a computational alternative that accomplishes the same thing.

Summary: Production Systems as Frameworks for Cognitive Developmental Theory

The production-system approach to theory building in cognitive development rests on three fundamental premises:

1. The human information-processing system architecture is isomorphic to a production-system architecture. This premise derives from observations about similarities in terms of both structural organization and behavioral properties. Structurally, production systems provide a plausible characterization of the relations between long-term memory and working memory, and about the interaction between procedural and declarative knowledge. Behaviorally, strong analogies can be seen between humans and production systems with respect to their abilities to mix goal-driven and event-driven processes, and with their tendency to process information in parallel at the recognition level and serially at higher cognitive levels.
2. Change is a fundamental aspect of intelligence; we cannot say that we fully understand cognition until we have a model that accounts for its development. The first 20 years of information-processing psychology devoted scant attention to the problems of how to represent change processes, other than to place them on an agenda for future work. Indeed, almost all of the information-processing approaches to developmental issues followed the two-step strategy outlined in the Simon quotation that opened this chapter: First construct the performance model, and then follow it with a change model that operates on the performance model. In recent years, as researchers have begun to work seriously on the change process, they have begun to formulate models that inextricably link performance and change. Self-modifying production systems are one such example of this linkage.
3. All information-processing-system architectures, whether human or artificial, must obey certain constraints in order to facilitate change. It is these constraints that give rise to the seemingly complex particulars of individual production-system architectures. Thus, an understanding of production-system models of change is a step toward understanding the nature of human development and learning.

CONNECTIONIST SYSTEMS

In this section we examine work conducted from the connectionist perspective. Because both production system modelers and connectionists are pursuing common goals, there are many points where their pathways converge. Both approaches rely heavily on computational modeling. Both approaches understand the importance of matching theory to data. Both perspectives have come to understand the importance of emergent properties in understanding transition mechanisms. Since the final understanding of transition mechanisms may well require insights from both perspectives, it makes little sense to advance strong claims for superiority of one approach over the other. Rather, we need to understand why researchers are currently exploring different paths, invoking different incantations, and wielding different computational weapons. To do this, we need to better understand the differences in the goals and constraints assumed by the two approaches.

We start with a brief description of the basic features of connectionist models. Then we address a few important aspects of connectionism that distinguish it from production system approaches. One basic distinction comes from the fact, noted earlier, that production systems take the symbol as their basic building block, while connectionist systems take a "sub-symbolic" perspective. Although we have reserved most of the "compare and contrast" discussion in this chapter for the final section, it is important to treat this distinction at the outset of our presentation of connectionist models. Following that discussion we turn to a review of actual work conducted in the connectionist framework.

Basic Principles of Neural Networks

Connectionist models are implemented in terms of artificial neural networks. Neural networks that are able to learn from input are known as "adaptive neural networks." In practice, all current neural network frameworks are based on adaptive neural networks. The architecture of an adaptive neural network can be specified in terms of eight design features:

1. *Units.* The basic components of the network are a number of simple elements called variously neurons, units, cells, or nodes. In Figure 13.3, the units are labeled with letters such as "x_1."
2. *Connections.* Neurons or pools of neurons are connected by a set of pathways which are variously called connections, links, pathways, or arcs. In most models, these connections are unidirectional, going from a "sending" unit to a "receiving" unit. This unidirectionality assumption corresponds to the fact that neural connections also operate in only one direction. The only information conveyed across connections is activation information. No signals or codes are passed. In Figure 13.3, the connection between units x_1 and y_1 is marked with a thick line.

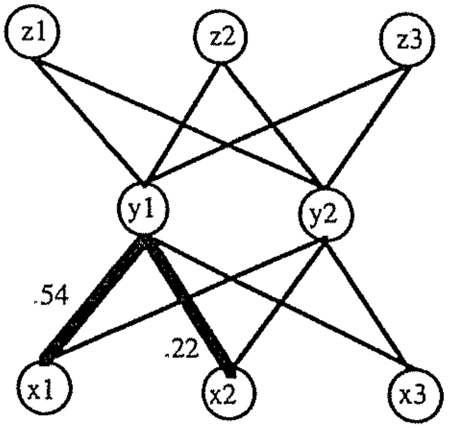

Figure 13.3 The general shape of a neural network.

3. *Patterns of connectivity.* Neurons are typically grouped into pools or layers. Connections can operate within or between layers. In some models, there are no within-layer connections; in others all units in a given layer are interconnected. Units or layers can be further divided into three classes:
   (a) *Input units* which represent signals from earlier networks. These are marked as x units in Figure 13.3.
   (b) *Output units* which represent the choices or decisions made by the network. These are marked as z units in Figure 13.3.
   (c) *Hidden units* which represent additional units juxtaposed between input and output for the purposes of computing more complex, nonlinear relations. These are marked as y units in Figure 13.3.

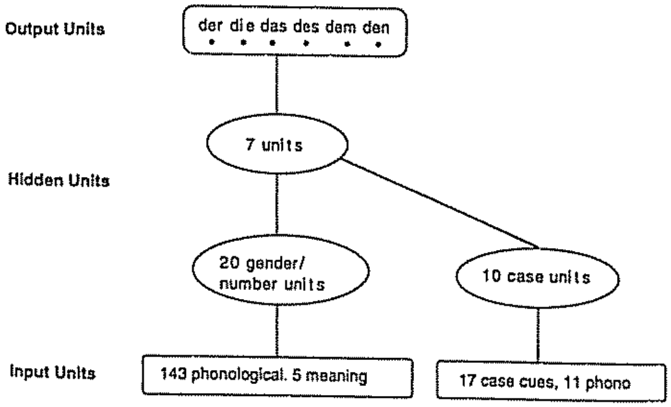

Figure 13.4 A network for learning the use of the German definite article. Based on "Language learning: Cues or rules?" by B. J. MacWhinney, J. Leinbach, R. Taraban,& J. L. McDonald, 1989, *Journal of Memory and Language, 28*, pp. 255–277. Copyright © 1989 by Academic Press, Inc. Reprinted with permission.

4. *Weights.* Each connection has numerical weight that is designed to represent the degree to which it can convey activation from the sending unit to the receiving unit. Learning is achieved by changing the weights on connections. For example, the weight on the connection between x_1 and y_1 is given as .54 in Figure 13.3.
5. *Net inputs.* The total amount of input from a sending neuron to a receiving neuron is determined by multiplying the weights on each connection to the receiving unit times the activation of the sending neuron. This net input to the receiving unit is the sum of all such inputs from sending neurons. In Figure 13.3, the net input to y_1 is .76, if we assume that the activation of x_1 and x_2 are both at 1 and the x_1y_1 weight is .54 and the x_2y_1 weight is .22.
6. *Activation functions.* Each unit has a level of activation. These activation levels can vary continuously between 0 and 1. In order to determine a new activation level, activation functions are applied to the net input. Functions that "squash" high values can be used to make sure that all new activations stay in the range of 0 to 1.
7. *Thresholds and biases.* Although activations can take on any value between 0 and 1, often thresholds and bias functions are used to force units to be either fully on or fully off.
8. *A learning rule.* The basic goal of training is to bring the neural net into a state where it can take a given input and produce the correct output. To do this, a learning rule is used to change the weights on the connections. *Supervised* learning rules need to rely on the presence of a target output as the model for this changing of weights. *Unsupervised* learning rules do not rely on targets and correction, but use the structure of the input as their guide to learning.

All connectionist networks share this common language of units, connections, weights, and learning rules. However, architectures differ markedly both in their detailed patterns of connectivity and in the specific rules used for activation and learning. For excellent, readable introductions to the theory and practice of neural network modeling, the reader may wish to consult Bechtel and Abrahamsen (1991) or Fausett (1994). For a mathematically more advanced treatment, see Hertz, Krogh, and Palmer (1991).

To illustrate how connectionist networks can be used to study cognitive development, let us take as an example the model of German gender learning developed by MacWhinney, Leinbach, Taraban, and McDonald (1989). This model was designed to explain how German children learn how to select one of the six different forms of the German definite article. In English we have a single word "the" to express definiteness. In German, the same idea can be expressed by *der, die, das, des, dem,* or *den.* Which of the six forms of the article should be used to modify a given noun in German depends on three additional features of the noun: its gender (masculine, feminine, or neuter), its number (singular or plural), and its role within the sentence (subject, possessor, direct object, prepositional object, or indirect object). To make matters worse, assignment of nouns to gender categories is often quite nonintuitive. For example, the word for fork is feminine, the word for spoon is masculine, and the word for knife is neuter. Acquiring this system of arbitrary gender assignments is particularly difficult for adult second language learners. Mark Twain expressed his own consternation at this aspect of German in a treatise entitled "The aweful German language" (Twain, 1935) in which he accuses the language of unfairness in assigning pretty young girls to the neuter gender, while allowing the sun to be feminine and the moon masculine. Along a similar vein, Maratsos and Chalkley (1980) argued that, since neither semantic nor phonological cues can predict which article accompanies a given noun in German, children could not learn the language by relying on simple surface cues.

Although these relations are indeed complex, MacWhinney et al. show that it is possible to construct a connectionist network that learns the German system from the available cues. This model, like most current connectionist models, involves a level of input units, a level of hidden units, and a level of output units (Figure 13.4). Each of these levels or layers contains a number of discrete units or nodes. For example, in the MacWhinney et al. model, the 35 units within the input level represent features of the noun that is to be modified by the article. Each of the two hidden unit levels includes multiple units that represent combinations of these input-level features. The six output units represent the six articles in the German language that correspond to the word the in English.

As noted, a central feature of such connectionist models is the very large number of connections among processing units. As shown in Figure 13.4, each input-level unit is connected to first-level hidden units; each first-level hidden unit is connected to second-level hidden units; and each second-level hidden unit is connected to each of the six output units. None of these hundreds of individual node-to-node connections are illustrated in Figure 13.4, since graphing each individual connection would lead to a blurred pattern of connecting lines. Instead a single line is used to stand in place of a fully interconnected pattern between levels. Learning is achieved by repetitive cycling through three steps. First, the system is presented with an input pattern that turns on some, but not all of the input units. In this case, the pattern is a set of sound features for the noun being used. Second, the activations of these units send activations through the hidden units and on to the output units. Third, the state of the output units is compared to the correct target and, if it does not match the target, the weights in the network are adjusted so that connections that suggested the correct answer are strengthened and connections that suggested the wrong answer are weakened.

MacWhinney et al. tested this system's ability to master the German article system by repeatedly presenting 102 common German nouns to the system. Frequency of presentation of each noun was proportional to the frequency with which the nouns are used in German. The job of the network was to choose which article to use with each noun in each particular context. After it did this, the correct answer was presented, and the simulation adjusted connection strengths so as to optimize its accuracy in the future.

After training was finished, the network was able to chose the correct article for 98% of the nouns in the original set. The ability to learn the input set is not a demonstration of true learning, since the network may have simply memorized each presented form by rote. However, when the simulation was presented with a previously encountered noun in a novel context, it chose the correct article on 92% of trials, despite the noun's often taking a different article in the new context than it had in the previously encountered ones. This type of cross-paradigm generalization is clear evidence that the network went far beyond rote memorization during the training phase. In addition, the simulation was able to generalize its internalized knowledge to entirely novel nouns. The 48 most frequent nouns in German that had not been included in the original input set were presented in a variety of sentence contexts. On this completely novel set, the simulation chose the correct article from the six possibilities on 61% of trials, versus 17% expected by chance. Thus, the system's learning mechanism, together with its representation of the noun's phonological and semantic properties and the context, produced a good guess about what article would accompany a given noun, even when the noun was entirely unfamiliar.

The network's learning paralleled children's learning in a number of ways. Like real German-speaking children, the network tended to overuse the articles that accompany feminine nouns. The reason for this is that the feminine forms of the article have a high frequency because they are used both for feminines and for plurals of all genders. The simulation also showed the same type of overgeneralization patterns that are often interpreted as reflecting rule use when they occur in children's language. For example, although the noun *Kleid* (which means clothing) is neuter, the simulation used the initial "kl" sound of the noun to conclude that it was masculine. Because of this, it invariably chose the article that would accompany the noun if it were masculine. Further, the same article-noun combinations that are the most difficult for children proved to be the most difficult for the simulation to learn and to generalize to on the basis of previously learned examples.

How was the simulation able to produce such generalization and rule-like behavior without any specific rules? The basic mechanism involved adjusting connection strengths between input, hidden, and output units to reflect the frequency with which combinations of features of nouns were associated with each article. Although no single feature can predict which article would be used, various complex combinations of phonological, semantic, and contextual cues allow quite accurate prediction of which articles should be chosen. This ability to extract complex, interacting patterns of cues is a particular characteristic of the type of connectionist algorithm, known as back-propagation, that was used in the MacWhinney et al. simulations. What makes the connectionist account for problems of this type particularly appealing is the fact that an equally powerful set of production system rules for German article selection would be quite complex (Mugdan, 1977) and learning of this complex set of rules would be a challenge in itself.

Connectionist Constraints on Computational Models

As we pointed out earlier, theoretical claims regarding production-system models do not extend to the underlying architecture of the computer on which they run. However, production systems have the potential to embody the same computational power as the von Neumann serial computer. Such models only become plausible as theories of human cognition when additional constraints are added, such as the size of working memory, the total amount of activation, and so on. However, some connectionists have not been satisfied with this analysis of the relation between von Neumann machines and human cognition. Instead, they have argued that the very nature of the underlying neural system yields emergent properties that are quite different from those implicit in production-system architectures. In particular, adaptive neural network models (Grossberg, 1987; Hopfield, 1982; Kohonen, 1982) deliberately limit this descriptive power of their models by imposing two stringent limitations on their computational models: a prohibition against symbol passing and an insistence on self-organization rather than hand wiring. We will describe each of these constraints below.

Thou Shalt Not Pass Symbols

The brain is not constructed like a standard digital computer. The crucial difference between the two machines lies in the structure of memory storage and access (Kanerva, 1993). In the random-access memory of a standard digital computer (von Neumann, 1956), there are a series of hard locations, each of which can store a single "word" of data. The size of the memory depends on the length of the word of data. Because the computer is built out of highly reliable electrical components, the integrity of each memory location can be guaranteed. Neural hardware is made out of noisy, unstable components and no such guarantees can be issued. To compensate for the lower reliability of individual components, the brain relies on massive parallelism and distributed memory encodings. In the type of neural memory that appears to be implemented in the cerebellum (Albus, 1981; Marr, 1969), the address space is huge and sparse. Because the system cannot rely on locating individual hard addresses at the site of individual neurons (Kanerva, 1993), it must perform retrieval by locating addresses in the general vicinity of the stored memory. These addresses are called "soft" memory addresses, since they refer not to a single location, but to a general position in address space. The address space has a huge number of dimensions; but, because it is so sparsely populated, retrieval of memories does not require the exact determination of hard addresses.

An alternative method for passing symbols between neurons would view individual neurons as separate processing units capable of sending and receiving signals. But we know that the signals sent and received by neurons are entirely limited in shape. Neurons do not send Morse code down axons, symbols do not run across synapses, and brain waves do not pass phrase structures. In general, the brain provides no obvious support for the symbol passing architecture that provides the power underlying the von Neumann machine.

Instead, computation in the brain appears to rely ultimately on the formation of redundant connections between individual neurons.

The ways in which the brain has adapted to these limitations are not yet fully understood. The cerebellar addressing system is probably only one of several neural memory systems that use soft addresses and other storage techniques. We know that the hippocampus is also involved in aspects of memory storage (Schmajuk & DiCarlo, 1992) and it appears that its role may involve techniques involving data compression. There are also various rehearsal pathways designed to implement the learning of verbal material (Gathercole & Baddeley, 1993; Gupta & MacWhinney, 1994, 1996). Our emerging understanding of the various memory systems of the brain points to a complex interaction between cortex, thalamus, hippocampus, cerebellum, and other brain structures that work both on line and during sleep to facilitate storage, learning, and retrieval of memories. All of this work is done in ways that circumvent the limitations on symbol passing imposed by the biological structure of neurons.

Thou Shalt Not Hand-Wire

By itself, the requirement that computation be performed locally without symbol passing or homunculi is not enough to fully constrain the descriptive power of our models. One could still hand-wire a neural network to perform a specific function or to model a particular behavior. In neural networks, hand-wiring can be accomplished by creating a little program or homunculus that gets inside the network and sets weights on individual links between nodes. For example, we could hand-wire an animal category by linking nodes labeled "cat," "dog," and "tiger" to a hand-coded node labeled "animal." By detailed weight setting and the use of gating and polling neurons, virtually any function can be wired into a neural network (Hertz et al., 1991). An early example of a fully hand-wired neural network was Lamb's (1966) stratificational grammar. More recently, we have seen hand-wired neural networks in areas such as interactive activation models of reading (McClelland & Rumelhart, 1981), speech errors (Dell, 1986; MacWhinney & Anderson, 1986; Stemberger, 1985), ambiguity resolution (Cottrell, 1985), and lexical activation (Marslen-Wilson, 1987). Although these networks fit within the general framework of connectionist models, the fact that they are constructed through hand-wiring makes them less interesting as developmental models.

Certain "hybrid" models move the process of hand-wiring away from the network level onto an alternative symbolic level. This "implementational" approach to hand-wiring spares the modeler the tedium of hand-wiring by running the wiring procedure off symbolic templates. For example, Touretzky (1990) has shown that there are techniques for bottling the full power of a LISP-based production-system architecture into a neural net. These demonstrations are important because they show how difficult it is to control excessive modeling power.

Ideally, we want to match the constraint against symbol passing with the requirement that networks be *self-organizing*. We want to make sure that specific representations are not hand-wired and that the connections between units are developed on the basis of automatic learning procedures. Although we will always be forced to "label" our input nodes and output nodes, we want our labelling systems to be general across problems and not hand-crafted anew for each particular problem. Rather, we want to use general forms of representation that lead to robust and emergent learning without recourse to hand-wiring. It is the emergent, self-organizing properties of neural networks that make them particularly interesting to the developmental psychologist. Such models can display further interesting and important properties, such as stage transitions (Shultz, Schmidt, Buckingham, & Mareschal, 1995), category leakage (McClelland & Kawamoto, 1986), graceful degradation (Harley & MacAndrew, 1992; Hinton & Shallice, 1991; Marchman, 1992), and property emergence (MacWhinney et al., 1989).

Alternative Network Architectures

One of the principal goals of connectionist theory over the last thirty years has been the exploration of the properties of competing network architectures. In this section we will review the most important network architectures with an eye toward understanding the types of developmental processes for which each might be most relevant. There is a great deal of evidence to suggest that no single architecture is ideal for all purposes and that the human brain probably uses different patterns of neural connectivity to solve different cognitive problems.

Perceptrons

In the late 1950s, researchers (e.g., Rosenblatt, 1959; Block, 1962; Widrow & Hoff, 1960) explored the properties of a simple connectionist model called a perceptron. This model connected a series of input units to one or more output units using simple unidirectional connections. The weights in the network were trained using an algorithm called the perceptron learning rule. The perceptron learning rule comes along with the rather attractive guarantee that, if a perceptron can be configured to solve a problem, the algorithm will succeed in finding the solution. The rub is that it often turns out that perceptrons cannot solve even very simple problems. For example, Minsky and Papert (1969) showed that perceptrons can encode a relation such as "black and tall," but not a relation such as "black but not tall." The problem with perceptrons is not with the learning rule, but with the strength of the basic computational mechanism. Today, perceptrons are only of historical interest.

Pattern Associators and Backpropagation

The successors to the perceptron are the pattern associators, and there are dozens of pattern associator architectures. Typically, these devices are designed as models of retrieval in human memory. They rely for their power on the holographic quality of neural networks which are able to retrieve stored patterns through vector manipulations. For example, a pattern associator should be able to take the sound /bal/ and retrieve the spelling B-A-L-L or it can take the smell of a rose and retrieve the vision of the thorns of the rose. Networks of this type are often trained using the delta rule or the extended delta rule. These rules compare the network's output patterns against some target signal and make weight adjustments to bring the network into line with the target.

The backpropagation architecture (Werbos, 1974) achieves additional computational power by adding an additional level of units between the input and output layers. These additional units are called "hidden units" because they have no direct connection to either the input or the output. Networks using backpropagation with hidden units and the delta rule can solve many types of problems that are difficult for simpler machines such as the perceptron. In fact, most current work in computational modeling of developmental phenomena makes use of the backpropagation framework. This single, simply characterized algorithm has demonstrated an ability to learn a wide variety of subtle patterns in the data.

Despite the proven success of backpropagation, there are several crucial problems that arise when we try to use this single architecture as an account for all aspects of cognitive and linguistic development. Each of the problems encountered by backpropagation has served as a stimulus to the development of interesting alternative frameworks. One basic problem that arises immediately as we try to match the backpropagation algorithm up to the brain is the fact that backpropagation assumes that connections which fire in a feed-forward fashion can also be trained in a feedbackward direction. However, we know that real neurons fire in only one direction and that this type of backwards training is not neurologically plausible. However, as Fausett (1990) shows, one can devise backpropagation networks that can be trained in a unidirectional and local manner by adding additional arrays of controlling units.

The study of the actual mechanics of weight changing in neural networks is very much the province of the cellular neurophysiologist. In this area, there is increasing evidence emphasizing the extent to which the neuron can compute complex functions. Hebb (1949) suggested that learning occurs when two cells fire simultaneously and the output of the postsynaptic cell functions to strengthen the firing of the synapse connecting the two cells. Although work by Kandel and Hawkins (1992) with the sea slug supports aspects of the Hebbian model of learning, Alkon and colleagues (1993) have found computationally more complex learning in higher organisms such as rabbits and rats. This non-Hebbian learning takes place locally on small areas of the dendritic cell membrane. Alkon has implemented a network model called Dystal that faithfully mimics these aspects of membrane activity and also works well as a connectionist pattern associator.

Networks That Deal with Time

In the standard backpropagation framework, processing is idealized as occurring at a single moment in time. This idealization may make sense for processes that are extremely brief or for decisions in which many factors are being weighed without time constraints. However, for problems such as word recognition, sentence production, seriation, and speeded chess playing, temporal components are crucial components of the task. One network architecture that deals with this problem is a variation on back propagation developed by Jordan (1986) and Elman (1990). This variation takes the standard three-layer architecture of pools A, B, and C shown in Figure 13.5 and adds a fourth input pool D of context units which has recurrent connections to pool B. Because of the recurrent or bidirectional connections between B and D, this architecture is known as *recurrent backpropagation*.

A recurrent backpropagation network encodes changes over time by storing information regarding previous states in the pool of units labeled as D. Consider how the network deals with the processing of a sentence such as "Mommy loves Daddy." When the first word comes in, pool C is activated and this activation is passed on to pool B and then

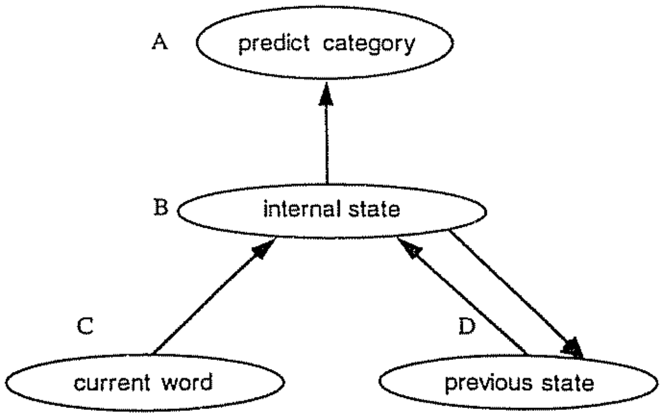

Figure 13.5 A recurrent backpropagation network.

pools A and D. The complete state of pool B at Time 1 is stored in pool D. The activation levels in pool D are preserved, while pools A, B, and C are set back to zero. At time 2 the networks hears the word "love" and a new pattern of activations is established on pool C. These activations are passed on to pools B, C, and D. However, because pool D has stored activations from the previous word, the new state is blended with the old state and pool C comes to represent aspects of both "Mommy" and "love."

Processing in a network of this type involves more than just storage of a superficial sequence of words or sounds. For example, in the simulations of sentence processing developed by Elman (1993), the output units are trained to predict the identity of the next word. In order to perform in this task, the network needs to implicitly extract part-of-speech information from syntactic co-occurrence patterns. Alternatively, the output units can be used to represent comprehension decisions, as in the model of MacWhinney (1996). In that model, part-of-speech information is assumed and the goal of the model is to select the agent and the patient using a variety of grammatical and pragmatic cues.

Another method for dealing with temporal ordering was developed by Grossberg (1978). In this system, linear ordering of elements such as the phonemes in a word is controlled by cluster units which sit above the component phoneme units and control their ordering as what Grossberg calls an "avalanche." The Elman and Grossberg systems are designed for markedly different problems. Grossberg's system works well for the learning of invariant serial orderings such as those found in lexical phonology and Elman's system is more appropriate for the learning of flexible, variant patterns of serial ordering, such as those found in syntax. It would not be surprising to find that other problems in serial ordering required still other network architectures.

Avoiding Catastrophes. A serious limitation of the backpropagation algorithm is its tendency toward developmental instability. A backpropagation network trained on one set of inputs can undergo a process of "catastrophic interference" (McCloskey & Cohen, 1989) when the input corpus is shifted to a markedly different structure. The problem of catastrophic interference shows up clearly when a network is trained with one language (L1) and then suddenly switched to dealing with input from a second language (L2). What happens is that learning of L2 wipes out knowledge of L1 (MacWhinney, 1996). Of course, no such catastrophic interference occurs in real life. When we learn a second language in real life, our knowledge of our first language remains firm.

Catastrophic interference occurs in backpropagation networks because new memories tend to overwrite old memories. One class of solutions tries to address this problem by making minor changes to backpropagation. This can be done by making weight changes only for novel aspects of the input (Kortge, 1990), hand-tuning the input corpus to avoid sudden changes (Hetherington & Seidenberg, 1989), localizing the receptive fields for units (Kruschke, 1992), or adding units with different learning rates (Hinton & Plaut, 1987). Although these solutions solve the problem of catastrophic interference, they often force us to make overly restrictive assumptions about the possible distributions of cues in the environment.

Localized Memories. A more general approach to the problem of catastrophic interference and other forms of crosstalk focuses on the role of neuronal topology in controlling neuronal recruitment and memory development. In topological models, units are more specifically devoted to specific memories, interactions between memories tend to be confined to local areas, and major shifts in the character of the input do not overwrite these localized memories.

Kanerva's Sparse Distributed Memory (SDM) is one such topological approach. The SDM model allows for one-shot storage of new memories without crosstalk. However, memories must be stored at several neighboring locations to guarantee consistent retrieval. A similar framework has been proposed by Read, Nenov, and Halgren (1995) on the basis of Gardner-Meadwin's (1976) model of hippocampal functioning.

The idea of encoding memories through topological organization in the brain is further elaborated in the self-organizing feature map (SOFM) approach developed by Kohonen (1982) and Miikkulainen (1990). Self-organizing feature maps use an unsupervised, competitive learning algorithm. All input units are connected to cluster units which are organized in a two-dimensional topological grid (see Figure 13.6), which is actually a compressed representation of a multidimensional space. When an input is presented, the cluster unit that responds most strongly becomes the winner. The winning unit then decrements the units that are just outside its immediate neighborhood so that they are less likely to respond to a similar input when it is next presented. The pattern of inhibition follows the "Mexican hat" format found in cells of the visual cortex. In this way, two units that initially respond to the same set of inputs start to pull away from each other. As this process continues, the radius for each unit decreases and its specificity increases. MacWhinney (1996) found that a self-organizing feature map of 10,000 units was able to learn an array of 6,000 words with 99% accuracy. Thus, it seems that the SOFM architecture is well-suited for the learning of arbitrary associations such as words.

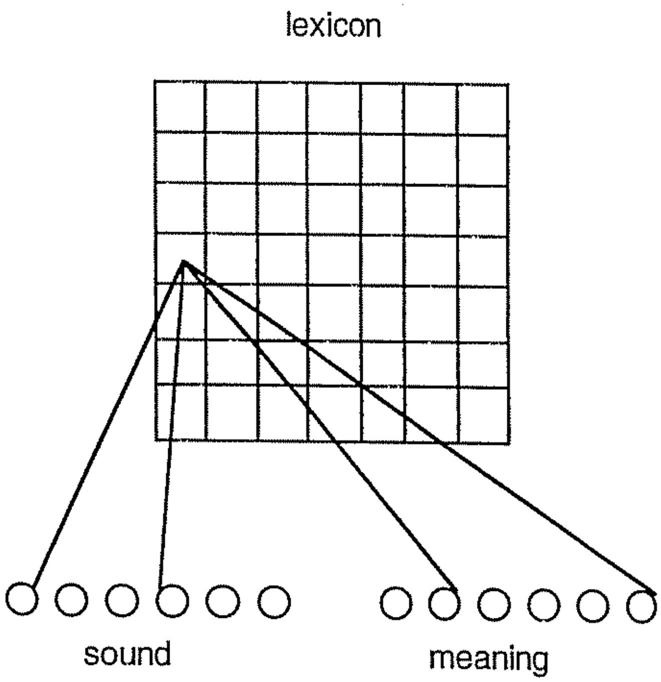

Figure 13.6 A self-organizing feature map illustrating connections for one lexical item.

The success of feature maps in the learning of arbitrary associations, such as the sound-meaning associations involved in words, stands in marked contrast to the problems that backpropagation networks have with the same task. The backpropagation architecture is designed to detect patterns, rather than to encode arbitrary associations. When a backpropagation network is trained with a long list of English words, it will lose its ability to acquire new words after learning the first 700 words or so. Adding more hidden units to the network does not help at this point, since the limitation seems to be in the basic resolution of the weight space. The reason that backpropagation reaches saturation for learning new words is not because of the shortage of nodes, but because of problems with the basic algorithm. Backpropagation uses hidden units not as individual address spaces for individual lexical items, but as pattern detectors that search for commonalities between words. However, because words are really arbitrary associations between sounds and meanings, backpropagation is frustrated in its attempt to pick up meaningful or useful patterns. The SOFM architecture, on the other hand, can be used to simply throw a large number of only weakly associated memories onto a large feature map. As MacWhinney (1996) has found, feature maps and sparse distributed maps can learn items up to the size of the feature map. In this regard, they seem better suited to the task of lexical learning than does an architecture such as backpropagation.

Networks That Grow. In addition to the crosstalk problem that lies at the root of catastrophic interference, backpropagation networks also suffer from a problem with commitments to local minima during early phases of training. These networks tend to isolate the major patterns in the input early on and are often incapable of picking up secondary strategies that conflict with the basic patterns in the input. One way of solving this problem is to force the network to "start small." By giving the network only minimal resources at first and allowing it to recruit new resources when the problem becomes more difficult, it is possible to force the network to treat basic statistical regularities as fundamental, while still learning higher-order regularities later.

Within the backpropagation framework, there have been quite a few recent proposals about how to add new units during learning (Azimi-Sadjadi, Sheedvash, & Trujillo, 1993; Fahlman & Lebiére, 1990; Frean, 1990; Hirose, Yamashita, & Hijiya, 1991; Kadirkamanathan & Niranjan, 1993; Platt, 1991; Wynne-Jones, 1993). One of these models is the "cascade correlation" approach of Fahlman and Lebiére (1990) which adds units when error reduction is not otherwise possible. The network begins its existence with only input and output units and no hidden units. In this form, it is equivalent to a perceptron. During training, new hidden units are added to the net in an effort to continually reduce the error in the output. As we will see below, this expansion of computational space through recruitment allows cascade-correlation networks to solve developmental problems that stymie standard backpropagation networks.

The idea of adding new units to networks to increase their computational capacity can be found in many frameworks. Within the framework of Adaptive Resonance Theory or ART (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992; Carpenter, Grossberg, & Reynolds, 1991; Grossberg, 1987), recruitment is a basic part of network functioning. For example, Grossberg (1987) adds new units to a network when no current unit matches a new input within a certain level of tolerance. Blackmore and Miikkulainen (1993) present a self-organizing feature map (SOFM) approach to incremental grid growing that allows for the expansion of a feature map to correct for errors in the compression of high dimensional feature space onto the two-dimensional topological grid.

A crucial insight incorporated in the various recruitment models is the idea that, by starting off with minimal computational resources, the learning system is forced to deal first with the most general patterns in the data. In effect, the system can only deal with the first of the various factors that can be extracted by principal components analysis (PCA). Once this first factor is learned, the network finds that there is still some residual error and it recruits new units to extract additional regularities. However, these new units are then largely dedicated to a second aspect of the problem. In this way, the network comes closer to modeling the type of stage-like learning we see in the child.

Recruitment versus Deletion

Models that rely on the recruitment of new neurons have been criticized on the grounds that they go against facts of developmental neurobiology. We know that new neurons are not added after birth. In fact, development is more characterized by neuronal loss than by neuronal addition. Some theorists have seized on this fact to argue that, like the immune system, neural development works by generating a vast array of potential cognitive structures which are then weeded out during development (Changeux & Danchin, 1976; Edelman, 1987; Jerne, 1967). Extending the analysis to issues in human development, Siegler (1989), Piatelli-Palmarini (1989), and Campbell (1960) have argued for the importance of "blind variation and selective retention" in creative thought and cognitive development.

Recent work calls into question some of the assumptions of this selectionist approach. Although it is true that there is a rapid loss of both cells and dendritic branches during the first months of life, the period of loss does not continue through development. Reassessing earlier claims about ongoing losses in synaptic density, Bourgeois, Goldman-Rakic, and Rakic (1994) have found ongoing synaptogenesis in prefrontal cortex during development. These findings match up with reports of increased volume of frontal cortex during development (Dekaban & Sadowsky, 1978; Jernigan et al., 1991) which indicate that brain development is not fundamentally selectionist and that additional resources may well be recruited during learning. This is not to say that the actual mechanisms supporting recruitment have yet been identified, only that models that use recruitment cannot be excluded on neurological grounds.

CONNECTIONISM AND DEVELOPMENT

We now turn to an examination of connectionist models of specific developmental processes. We will first look at models of the various component processes in language development. Second, we will examine general issues in cognitive development and their realization in connectionist models of specific cognitive tasks. Third, we will look at models of additional issues in development, including motor development and early brain maturation.

Connectionism and Language Development

The learning of language is a complex process that extends over the course of many years and which relies on interplay between several complex cognitive processes. Some recent accounts of language learning (Lightfoot, 1989; Pinker, 1994) tend to focus rather exclusively on the learning of grammatical markings and syntax, but language learning is more than just the learning of a few rules of grammar. In fact, the child devotes far more attention to tasks such as word learning, concept acquisition, articulatory control, discourse structuring, and conversational maintenance. No symbolic or connectionist model has been formulated that can handle all levels of language learning, although MacWhinney (1978, 1982, 1987a, 1988) and Pinker (1984) offered initial sketches in the symbolic framework. In the next sections, we will look at current work in connectionist models with an eye to discerning the shape of a new, more detailed, synthetic approach.

Word Learning

Current research in sentence processing (Juliano & Tanenhaus, 1993; Trueswell & Tanenhaus, 1991, 1992) has stressed the importance of individual words as determiners of sentence-level processing. The central role of words in phonological development and auditory learning has been recognized for nearly two decades (Ferguson & Farwell, 1975). Even discourse processes and narrative structures are grounded on specific lexical constructions (Goldberg, 1995).

For symbolic models, lexical learning is a computationally trivial problem, since symbolic models have no trouble picking up arbitrary numbers of arbitrary associations. However, the symbolic view of word learning as mere association does not match up well with developmental data. We know that children can learn new words quickly on the basis of a single encounter (Carey, 1978; Dollaghan, Biber, & Campbell, 1995), but only a few new words can be picked up at the same time. If we present a child with dozens of new words at once, learning starts to fall apart. Moreover, the exact nature of the representation constructed by the child learner or the adult second language learner (Atkinson, 1975) is often heavily dependent on the context of presentation (Kay & Anglin, 1982).

Networks that use backpropagation and hidden units have exactly the opposite problem with modeling lexical learning. These networks typically cannot learn more than about 700 lexical forms. After this level, the hidden units are so fully invested in distinguishing phonological and semantic subtypes and their associations that there is simply no room for new words. Adding more hidden units doesn't solve this problem, since all the interconnections must be computed and eventually the learning algorithm bogs down.

The problems faced by backpropagation are not general to all network models. Building on earlier models from Grossberg (1978, 1987), Houghton (1990), and Burgess and Hitch (1992), Gupta and MacWhinney (1996) have constructed a lexical learning model that uses three hierarchically-ordered layers, as illustrated in Figure 13.7. The lowest layer is a set of phonological units for the sounds of a word. Above this layer, is a set of phonological chunks representing the various syllable patterns of the language. On the top is a programmable level that controls the order of syllables within the word. Each new word is learned as a new node on the top layer and a series of weights on the connections from this layer to lower layers. Gupta and MacWhinney show that this model does a good job of accounting for a wide variety of well-researched phenomena in the literature on word learning, immediate serial recall, interference effects, and rehearsal in both adults and children (Gathercole & Baddeley, 1993).

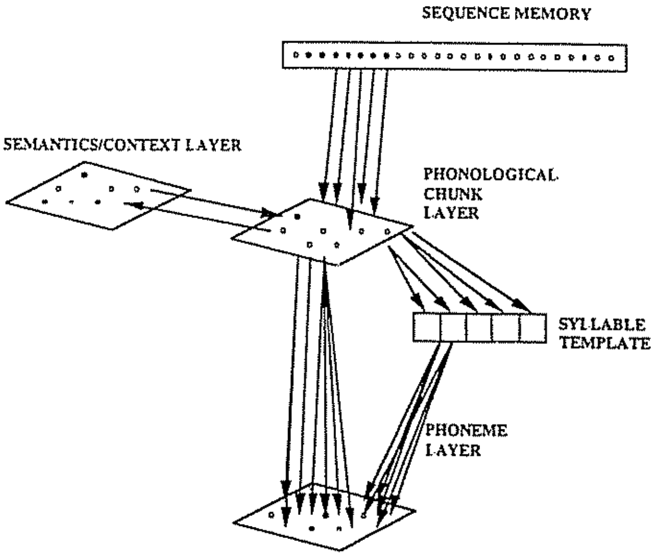

Figure 13.7 A lexical learning model using three hierarchically-ordered layers. From "Vocabulary acquisition and verbal short-term memory: Computation and neural bases," by P. Gupta & B. MacWhinney, 1996, *Brain and Language*. In Press. Copyright © 1996 by Academic Press, Inc. Reprinted with permission.

A major limitation of the Gupta and MacWhinney model is its reliance on a rigid fixed level of items for the top-level word nodes. Once the initial number of nodes has been used up, the system would need to recruit a new node for each new word. However, simply inserting a fresh node into the model for each new word requires us to make excessively strong assumptions about neuronal plasticity, while also failing to capture the ways in which new words interact with old lexical structures. As we noted earlier in our discussion of the model of MacWhinney (1996), by relying on a fuller address space of the type proposed by Kanerva or Miikkulainen, problems with the learning of new lexical items can be minimized.

Word Meaning

The task of simulating the semantic aspects of word learning is extremely challenging, because of the open-ended nature of meaning. Connectionist models have found two ways of dealing with this problem. One approach structures the world as a miniature perceptual system. This mini-world approach was developed first by Chauvin (1989) and echoed in a replication by Plunkett and Sinha (1992). The goal of the Chauvin model is to associate dot patterns with arbitrary labels. Learning is done using *autoassociation*. First the net is given an image and asked to activate a label, next it is given a label and asked to activate an image, and then it is given a trial with both image and label presented together. Plunkett and Sinha note that the cue validity of the label in their input corpus is higher than that of the image, since the label predicts a unique image, but an image does not predict a unique label. Unfortunately, it is difficult to see how a relation of this type can map onto the facts involved in real lexical structures where the opposite is usually the case.

Because the Chauvin/Plunkett-Sinha model uses backpropagation, it does not perform well in the basic lexical acquisition task. However, it does succeed in capturing some of the other phenomena associated with lexical learning. In particular, Plunkett and Sinha claim that their model captures these three phenomena:

1. *The prototype effect.* This effect replicates the original findings of Posner and Keele (1968; 1970) and has been observed in other connectionist models (McClelland & Rumelhart, 1985; Schyns, 1991), as well.
2. *The word learning spurt.* The model only starts producing labels after the first 20 epochs. At this point, Plunkett and Sinha view the onset of learning as similar to the "word spurt" that occurs in children. However, the cause of the delay in the model is the low validity of the image as a cue to the label and it is difficult to see how this structuring of lexical validities maps onto the real facts of lexical structures. Thus, it appears that the model is demonstrating the word learning spurt for the wrong reason.
3. *The superiority of comprehension over production.* Here, again, the reason for the superiority of comprehension is the higher cue validity of the label and it appears that the model is displaying the correct behavior for the wrong reasons.

Despite these limitations, the Chauvin/Plunkett and Sinha model serves as a useful starting point for thinking about comprehension-production relations in development (MacWhinney, 1990).

In another exploration within the mini-world framework, Schyns (1991) applied a Kohonen network to the task of learning three competing categories with prototype structures. The three categories were geometric patterns that were blurred by noise in order to create a prototype structure, although the actual prototypes were never displayed. The simulations showed that the network could acquire the patterns and demonstrate human-like categorization and naming behavior. When presented with a fourth new word that overlapped with one of the first three words, the system broke off some of the territory of the old referent to match up with the new name. Schyns interpreted this as evidence that the network was obeying the mutual exclusivity constraint of Markman (1989). However, the operation of his network can be understood even more clearly in terms of the forces of contrast and competition described by Clark (1987) and MacWhinney (1989).

An alternative approach to the development of word meaning focuses on the learning of small fields of real words. Three studies have been conducted to date. Shultz, Buckingham, and Oshima-Takane (1994) use the cascade-correlation algorithm to acquire the use of "you" and "me" from two types of input: child-directed speech and speech directed to a third party. The problem with learning the meaning of words like "you" and "me" is that the actual reference of the word is constantly changing. The best way to figure out the meanings of these words is to observe two other people using them. In this way, the child is able to see that "you" is used for the addressee and "me" for the speaker and that these words only have meaning in the context of the speaker-listener relation. Research by Oshima-Takane, Goodz, and Derevensky (in press) has shown that learning is faster when children are exposed to relatively more speech addressed to a third party, typically an older sibling, since this input makes the use of "you" clearer. By comparing input corpora with varying amounts of speech directed to a third party, Shultz et al. were able to model this effect.

Another study of meaning development by Li and MacWhinney (1996) used a standard backpropagation architecture to model the learning of reversive verbs that used the prefix "un-" as in untie or "dis-" as in disavow. The model succeeded in capturing the basic developmental stages reported by Bowerman (1982) and Clark, Carpenter, and Deutsch (1995) involving the production of errors such as unbreak or disbend. The network's performance was based on its internalization of what Whorf (1938; 1941) called the "cryptotype" for the reversive which involved a "covering, enclosing, and surface-attaching meaning" that is present in a word like untangle, but absent in a form such as unbreak. Whorf viewed this category as a prime example of the ways in which language reflects and possibly shapes thought.

The various models of learning word meaning we have discussed so far all treat meaning as if it were a fixed set of elements for any given word. However, nothing could be farther from the truth. Virtually every common word in our vocabulary has many alternative meanings and shades of meanings. In extreme cases, such as the verbs "put" and "run," the dictionary may list up to 70 alternative meanings. Typically, the choice of one meaning over another is determined by the other words in the sentence. For example, when we say that "the ball rolled over the table," we are thinking of the word over as meaning across. However, when we say that "Jim placed the snuffer over the candle," we are thinking of over as meaning covering. These competitions between the various alternative readings of words like "over" were discussed from a general connectionist perspective by MacWhinney (1989). Subsequently, an implemented connectionist model of the learning of the meanings of "over" by a network was developed by Harris (1990; 1994b). The Harris model is capable of taking new input test sentences of the type "the pin rolled over the table" and deciding on the basis of past learning that the meaning involved is across, rather than covering or above. It does this only on the basis of the co-occurrence patterns of the words involved, rather than on information from their individual semantics. Thus, it learns that combinations like ball, roll, and table tend to activate across without regard to facts such as knowing that balls are round and can roll or knowing that tables are flat and that rolling involves movement.

Inflectional Morphology

One of the most active areas of connectionist modeling has been the study of the child's learning of the ways in which words change when they are combined with grammatical markings such as suffixes or prefixes. These markings are called *inflections* and the system that governs the use of these inflections is called *inflectional morphology*. A simple case of inflectional learning is the system of patterns that help us choose to say "bent" instead of "bended" as the past tense of the verb bend. Inflectional learning is also involved in the learning of the correct form of the German definite article that we examined earlier. There are now well over thirty empirical studies and simulations investigating this topic from a connectionist perspective. The majority of work on this topic has examined the learning of English verb morphology with a particular focus on the English past tense. The goal of these models is the learning of irregular forms such as went or fell, along with regular past tense forms such as wanted and jumped. Other areas of interest include German noun declension, Dutch stress placement, and German participle formation. This work has examined six core issues:

1. *Cues versus rules.* The most central issue addressed in this research is whether or not one can model the learning of inflectional morphology without using formal rules. Pinker (1991) has argued that irregular forms are indeed produced by connectionist networks, but that regular forms are produced by a regular rule. However, Pinker's attempts to preserve a role for rules in human cognition runs into problems with the fact that even the most regular patterns or "rules" display phonological conditioning and patterns of gradience (Bybee, 1993) of the type that are easily captured in a connectionist network.
2. *Phonological representation.* Most current models of inflectional learning use a system for phonological representation like the one introduced by MacWhinney, Leinbach, Taraban, and MacDonald (1989). This system assigns each node a status on each of three coding systems. The first coding system indicates the position of the node in the syllable, the second indicates the position of the syllable in the word, and the third represents the presence or absence of a phonetic distinctive feature. Because this representational system relies on standard linguistic concepts, it addresses most of the concerns expressed by Pinker and Prince (1988) with earlier connectionist models of inflectional learning. An elaborated version of this same representational system can be found in Gasser (1991, 1992). Gasser's models emphasize the serial quality of morphological formations by relying on predictive recurrent networks. The system Gasser proposes uses three separate recurrent subnetworks for phonemic structure, syllabic structure, and metrical structure. On the top level, the three levels would be integrated in terms of lexical items. However, exactly how this integration of separate recurrent subnetworks should occur remains unclear, since Gasser never fully implemented his model.
3. *U-shaped learning.* A major shortcoming of nearly all connectionist models has been their inability to capture

the patterns of overgeneralization and recovery from overgeneralization known as u-shaped learning. Empirical work by Marcus et al (1992) has shown that strong u-shaped learning patterns occur only for some verbs and only for some children. The models of MacWhinney and Leinbach (1991) and Plunkett and Marchman (1991) showed levels of u-shaped learning in rough conformity with the patterns observed by Marcus et al. Moreover, Plunkett and Marchman showed that u-shaped learning levels could be affected by changes in the type and token frequencies of irregular verbs in the input.

4. *Rote learning of irregulars.* Although models like MacWhinney and Leinbach or Plunkett and Marchman succeed in demonstrating some u-shaped learning, this success is at least in part misleading. In order to correctly model the child's learning of inflectional morphology, models must go through a period of virtually error free learning of irregulars, followed by a period of learning of the first irregulars accompanied by the first overregularizations (Marcus et al., 1992). No current model consistently displays all of these features in exactly the right combination. MacWhinney (1996) has argued that models that rely exclusively on backpropagation will never be able to display the correct combination of developmental patterns and that a two-process connectionist approach will be needed. The basic process is one that learns new inflectional formations, both regular and irregular, by rote as items in self-organizing feature maps. The secondary process is a backpropagation network that uses the information inherent in feature maps to extract secondary productive generalizations.
5. *The role of semantic factors.* The first attempts to model morphological learning focused exclusively on the use of phonological features as both input and output. However, it is clear that the formation of past tense forms must also involve semantic factors. In English, the use of semantic information is associated with the irregular patterns of inflection. The idea is that, since we cannot access "went" by combining "go" and "-ed," it might be that we can access it directly by a semantic route. Of course, this idea is much like that underlying the dual-route theory. In German gender, the role of semantic information is much clearer. Köpcke and Zubin (Köpcke, 1994; Köpcke & Zubin, 1983, 1984; Zubin & Köpcke, 1981, 1986) have shown that a wide variety of both phonological and semantic factors are used in predicting the gender of German nouns and their plural. Some of the features involved include: alcoholic beverages, superordinates, inherent biological gender, gem stones, body parts, rivers inside Germany, and light versus heavy breezes. Simulations (Cottrell & Plunkett, 1991; Gupta & MacWhinney, 1992; MacWhinney, 1996) have integrated semantic and phonological information in various ways. However, a better understanding of the ways in which semantic factors interact during word formation will require a more extensive modeling of lexical items and semantic features.
6. *Extensions of irregular patterns to new words.* Extending earlier work by Bybee and Slobin (1982), Prasada and Pinker (1993) examined the abilities of native English speakers to form the past tense for nonsense words like plink, plup, or ploth. They found that, the further the word diverged from the standard phonotactic rules for English verbs, the more likely the subjects were to form the past tense by just attaching the regular "-ed" suffix. Ling and Marinov (1993) noted that the original verb-learning model developed by Rumelhart and McClelland (1987) failed to match these new empirical data, largely because of its tendency to overapply irregular patterns. To correct this problem, Ling and Marinov created a nonconnectionist symbolic pattern associator which did a better job modeling the Prasada and Pinker data. However, MacWhinney (1993) found that the network model of MacWhinney and Leinbach (1991) worked as well as Ling and Marinov's symbolic model in terms of matching up to the Prasada and Pinker generalization data.

Phonology

In the area of speech processing, connectionist models have been developed primarily as ways of simulating aspects of adult word recognition. The recurrent backpropagation architecture has been used in word recognition models developed by Norris (1994), Waibel, Hanazawa, Hinton, Shikano, and Lang (1988), and Watrous, Shastri, and Waibel (1987). Recently, Markey (1994) has developed a realistic physical representation of the young child's vocal apparatus and used it to model the development of phonetic and phonological skills. Markey's model is able to capture some of the basic aspects of early phonological development. Hopefully, we will soon see additional models that will allow us to better understand how much of early phonological development is determined by the articulatory apparatus and how much by the structure of the words being learned.

Reading

Sejnowski and Rosenberg (1988) presented an entertaining demonstration of a system called NETtalk that learned to read aloud. The system took as its input the orthographic representation of English words and was trained to produce computerized speech as its output. At first the network made only crude approximations to the sounds of the words and then moved phonologically closer and closer through training. In this regard, the network fails to actually capture the nature of early reading in the child where words are fully formed phonologically and the task is to extract enough cues to effect retrieval of the full word form (Simon, 1976; Simon & Simon, 1973). However, a positive aspect of the NETtalk model is its ability to extract local graphemic linear dependencies that a beginning reader might use to derive the sound of a word.

A more complete picture of the development of early reading skills was provided in a backpropagation model developed by Seidenberg and McClelland (1989). An important quality of this model is that it emphasizes the ways in which both regular spellings such as hint or mint can be controlled by the same computational mechanism that also controls irregular spellings such as pint. In the traditional symbolic approach (Marshall & Newcombe, 1973), a distinction is made between rote storage for irregular forms and pattern-based storage for regular forms. This distinction motivates a dual-process or dual-route approach to reading. Seidenberg and McClelland show that one can model the learning and usage of both regulars and irregulars in a single model with a single set of processes.

The Seidenberg-McClelland model has been challenged on empirical grounds (Behrman & Bub, 1992; Besner, Twilley, McCann, & Seergobin, 1990; Coltheart, Curtis, Atkins, & Haller, 1993). One problem with the model was its inability to acquire its own training set. However, by using a phonological input representation much like that developed by MacWhinney et al. (1989), Plaut, McClelland, Seidenberg, and Patterson (1995) were able to improve on the performance of the original model. A second problem with the model arose in connection with the modeling of data from neurological patients with deep dyslexia. For these patients, the model underestimated the sparing of high-frequency regular and irregular forms, as predicted by the dual-route model. Here, again, the revised coding system of Plaut et al. was able to improve on the performance of the original model.

In evaluating the status of the debate between single-route and dual-route accounts of reading and lexical processing, it is important to recognize that connectionist theory makes no specific commitment to the single-route concept. Moreover, it may be impossible to avoid some aspects of duality, even in the most homogeneous model. For example, Kawamoto (1993; Kawamoto & Zemblidge, 1992) has shown that subjects tend to produce incorrect pronunciations of irregulars more quickly than correct pronunciations. Thus, the pronunciation of pint to rhyme with hint is faster than the correct pronunciation of pint. Kawamoto models this effect using a ART-type model. At first, the large number of words with the regular "-int" shape activate a common pattern. If the subject produces a reading of the word at this time, it will be an error. A few milliseconds later, the slower connections to the irregular pronunciation start to dominate and the correct pronunciation will be produced. This is still a single mechanism, but the presence of two routes is simulated by contrasting pattern activations at different time points during the settling of the network.

Syntactic Classes

Psycholinguists working in the standard symbolic tradition (Chomsky, 1965; Fodor & Pylyshyn, 1988; Lachter & Bever, 1988) have pointed to the learning of syntax as a quintessential problem for connectionist approaches. One of the key abilities involved in the learning of syntax is the abstraction of syntactic classes or "parts of speech," such as nouns, verbs, or prepositions. In the theory of universal grammar, these categories are innately given. However, their actual realization differs so much from language to language that it makes sense to explore accounts that induce these categories from the input data. Elman (1993) has presented a connectionist model that does just this. The model relies on a recurrent architecture of the type presented in Figure 13.7 above. The training set for the model consists of dozens of simple English sentences such as "The big dog chased the girl." By examining the weight patterns on the hidden units in the fully trained model, Elman showed that the model was conducting implicit learning of the parts of speech. For example, after the word big in our example sentence, the model would be expecting to activate a noun. The model was also able to distinguish between subject and object relative structures, as in "the dog the cat chased ran" and "the dog that chased the cat ran."

Even more interestingly, Elman found that the network only learned to pick up these positional expectations when it began with a narrow perceptual window of two or three words. If the network started with too large a window, it could not focus on detection of the most basic determinants of syntactic positioning. Elman interpreted this contrast as underscoring the "importance of starting small." In many ways, this analysis is much like the one offered by Schultz for the importance of a learning algorithm that starts off with limited resources and only recruits new resources when it is unable to further reduce error.

Lexical Segmentation and Masking

In order to process sentences effectively, we need to be able to segment out words from the ongoing speech stream. Norris (1994) proposes a system called ShortList which uses a recurrent net of the type given in Figure 13.5 to process incoming phonemes left-to-right. A network of this type does fine with many words. However, it has trouble with words like catalog which have what Norris calls a "right context" problem. When processing the word catalog, a simple recurrent net would recognize the word cat and decide that this word had actually occurred, if it were not somehow forced to hold off and process further right context. In order to prevent this from happening, Norris suggests that there must be a short list of competitors that include words like cat, cattle, catalog, and others like them that will compete for full recognition of the input material. The ShortList implementation of this process uses a hand-wired word list. However, Miikkulainen (1993) has suggested that it would be possible to model this same process using self-organizing feature maps.

Once a word has been successfully detected, the sounds that activated it need to be masked out, in order to block multiple recognition of the same input by alternative competitors. Take a sentence like "I gave my cat a Miranda doll." Once the word cat has been selected, its component phonemes are "masked" in order to avoid the additional activation of the form "catamaran" on the basis of the string "cat a Miran." Once a word is fully recognized and its component sounds are masked, it must then begin to participate in higher level syntactic and semantic patterns. The exact nature of this conversion is not yet clear. There have been several suggestions regarding the nature of this short-term verbal memory:

1. As soon as words are linked together into conceptual clusters, they can be used to activate a unique underlying meaning that no longer requires verbal storage.
2. Before this linkage occurs, words may be retained in a phonological loop (Baddeley, 1986). This immediate rehearsal requires that words be present in a primarily articulatory form (Gupta & MacWhinney, 1994).
3. It is also possible that some additional mechanism operates on lexical items to encode their serial occurrence without reference to either meaning or sound. This could be done in terms of some additional episodic, possibly hippocampal, mechanism that stores activation levels of words prior to masking. A system of this type is close to the Competitive Queuing mechanism proposed first by Grossberg and then again by Houghton.

Further experimental work will be needed to decide which of these three mechanisms is involved at which points in the storage of short term verbal memories. However, there is already good (Gupta & MacWhinney, 1996) evidence that various neural mechanisms are available to support masking in the lexicon.

Bilingualism

The study of bilingualism and adult second language learning is a particularly promising and challenging area for connectionist research. Recent research in second language acquisition (Dechert & Raupach, 1989; Flege & Davidian, 1984; Hancin-Bhatt, 1994; Harrington, 1987; Johnson, 1989; MacWhinney, 1992; Odlin, 1989; Sasaki, 1994) has underscored the importance of transfer of first language skills to the learning of the second language. Because of its emphasis on pattern generalization, the backpropagation algorithm is well-suited to modeling transfer effects. In one of the first simulations designed to examine these issues, Gasser (1990) constructed an auto-associative network that used backpropagation training for the learning of basic word orders in second language learning. In one simulation, the network was first trained with a first language order of Subject-Verb (SV) and then exposed to a set of second language sentences with Verb-Subject (VS) order. In the other "mirror-image" simulation, the network began with VS in the first language and then shifted to SV in the second language. The network demonstrated a strong tendency to transfer the first language word order to the second language, particularly for words that were similar semantically. This type of lexically-based transfer for word order is exactly what one would expect for a strong pattern generalizing network. However, there is not yet any actual empirical data that would support the importance of this effect in real second language learning.

MacWhinney (1996) reports on unpublished work by Janice Johnson that adapts the architecture of the recurrent network shown in Figure 13.5 to the problem of second

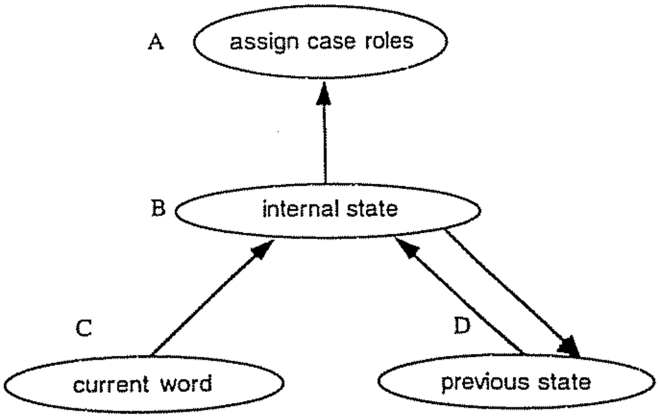

Figure 13.8 Architecture for the Johnson-MacWhinney model for second language learning.

language learning. The exact shape of the model is given again as Figure 13.8.

In these simulations, the input in pool C is a pattern that represents the status of the "current word" along the dimensions of animacy, number, case, agreement-marking, part-of-speech, and language. We assume that this information is available through the individual lexical item. Note that this highly structured form of the input differs radically from the raw word level input used by Elman (1993). Because of the highly structured shape of the input, this network performs much better than the Elman net as a sentence processor and interpreter. The task of the network is to assign the agent, object, and perspective roles to the correct words. In order to get these assignments right, the network must activate the correct output units in pool A. For example, the network can choose between activating a node that assigns the first noun as agent and a competing node that assigns the second noun as agent. Training involves the presentation to the network of sentences, one word at a time. For example, the input could be "the dog is chased by the cat." In this case, the network might begin by thinking that "the dog" is the agent. However, once the passive form of the verb is detected, the weight on this role assignment is decreased and the second noun is selected as agent instead. When the network is processing passive sentences, we find that it goes through an on-line reversal or "garden-path," first activating a choice of the first noun as agent and then reversing this activation to choose the second noun as the agent.

The network was trained initially with a wide variety of English sentence patterns. At the end of this initial training, it was performing well in the role assignment task for English. Then, the input was extended to include a full corpus of parallel sentences for Dutch. After a period of mixed training, the network then continued with Dutch-only training for a further period. The first basic finding of this research was that the exact weights of the various cues in the model matched up well with a large body of empirical research summarized in MacWhinney and Bates (1989). In particular, the model learned the basic English SVO pattern quickly and then continued to learn the VOS and OSV patterns found in adult speakers. The second finding was that, when learning Dutch, the model showed exactly the type of word order transfer effects reported by McDonald (1987, 1989) for the learning of Dutch by English speakers and the learning of English by Dutch speakers. Finally, the model also showed a clear tendency toward "catastrophic interference" if the period of mixed-language training was omitted. A more robust, general approach to the catastrophic interference problem in this network and others like it could be developed if the network were given a firmer grounding on the learning of syntactic patterns on the basis of generalization from particular lexical items, as we noted earlier.

Connectionism and Developmental Theory

Connectionism offers a fresh perspective on a variety of issues of ongoing concern to developmentalists, including the emergence of symbols and representations, the movement between developmental stages, and the role of nonlinearities in development.

Stages and Transitions

The simplest developing system is one that shows only one type of uniform change over time. For example, a falling ball undergoes only one type of transition during its downward movement. We can use Galileo's equation for acceleration to compute the distance traveled as a function of acceleration a and time t. For this simple, uniform system, we have a clear rule that allows us to predict the state of the system at each time t.

More complex systems can go through a series of stages during state transitions. For example, a drop of rain can begin as cloud vapor, form into a droplet, freeze into hail, fall to the ground, and then melt into slush. Each of these state transitions delimit specific stages in the life of the droplet. In the human child, stages of this sort abound. For example, after learning the first word, children spend several months slowly picking up a few additional words. Then, suddenly, we see a rapid growth in vocabulary that has been called the vocabulary "burst." The vocabulary burst does not emerge overnight, but builds over the course of several weeks. However, if we plot the size of the vocabulary on the y-axis and the child's age on the x-axis we will note a marked upward acceleration at the beginning of this period. Such changes indicate a stage-like quality in development.

Piaget has characterized the intellectual growth of the child in terms of four major epochs, each composed of several periods with some further divisions of the periods into subperiods. However, Piaget's characterization of these stages as invariant properties of human development is no longer widely accepted and few researchers are interested in developing simulations to account for the child's movement through the classical set of Piagetian stages. This is not to deny the reality of major qualitative changes in cognition as the child moves from infancy to adolescence. However, attempts to capture these changes across skill domains have not been successful. Because of this, connectionist models of stagelike transitions have tended to focus not on broad changes in cognition, but on local discontinuities within the development of specific skills. The areas that have been most closely investigated are the balance beam, velocity computation, and seriation.

Balance Beam

One of the clearest analyses of stage transitions in cognitive development is the case of the balance beam problem studied first by Piaget. Earlier we examined the production-system accounts for learning of the balance beam problem by Klahr and Siegler (1978). McClelland (1989, 1995) has noted that, although these production-system models provide a good description of the four balance beam rules discussed earlier, they tell us little about the forces that drive the child from one rule system to the next.

McClelland was able to construct a backpropagation model of the balance beam problem that used 20 input units. There were 10 positional units devoted to the 5 positions to the left of the fulcrum and the 5 positions to the right of the fulcrum. The 10 weight units were dedicated to represent the numbers of weights stacked up at a position with 5 units for the possible number of weights on the left and 5 units for the possible weights on the right. A given problem could be encoded with a total of 4 units turned on. For example, take a problem with 4 weights at a distance of 3 right and 5 weights at a distance of 2 left. The units turned on would then be 4-right-weight, 5-left-weight, 3-right-distance, and 2-left-distance. In order to bias the network toward reliance on the weight cue over the distance cue, McClelland included a large number of cases in which the distance cue was neutralized, thereby focusing the network's attention to the weight cue.

Using this type of representation, McClelland was able to model many aspects of the learning of this task. The network began with performance that relied on Rule 1 and moved on to learn Rule 2 and then Rule 3. It never acquired full use of Rule 4, but, McClelland argues, this is because some aspects of the use of Rule 4 in adults involve the application of full mathematical analysis. However, the network was able to capture aspects of the rather subtle "torque distance effect" detected in studies by Ferretti and Butterfield (1986) and Wilkening and Anderson (1982). These studies have shown that subjects perform best and most consistently on balance beam problems when it is clear perceptually that one side has an overwhelming combination of weight and distance in its favor. When the balance between the two sides is closer numerically, decisions are less consistent. Torque distance effects indicate that subjects are not simply applying an all-or-none rule, but are performing a type of cue-weighting that is much like that conducted inside a neural network.

Shultz, Schmidt, Buckingham, and Mareschal (1995) extended McClelland's model by using the cascade correlation variation of the backpropagation algorithm. Shultz et al. argue that static backpropagation networks with only a few hidden units can succeed at modeling the first stages of development, but are unable to reach higher levels of performance, because their weights become too closely tuned to solving the basic levels of the problem. This was true for McClelland's balance beam model, which learned Rules 1, 2, and aspects of 3, but was unable to learn Rule 4. However, using the cascade correlation framework, Shultz et al. were able to model successful learning of all four rules.

These models make two important points. First, both the McClelland and the Shultz et al. models show that connectionist models can provide good accounts of perceptual aspects of learning such as the torque distance effect. Second, Shultz's model shows that static networks that begin life with abundant numbers of extra hidden units may fail to perform the type of architectural decomposition of a problem space that is required for successful mastery. Models that start out small and are forced to recruit new units when they run out of steam are more likely to be able to focus first on the core of a problem and then add the details as elaborations of this core.

Other Physical Coordinations

In addition to their work on the balance beam problem, Shultz and his colleagues have developed connectionist models for three other types of physical coordinations. These include the learning of seriation, potency-resistance, and velocity-distance-time relations. Mareschal and Shultz (1993) developed a model of seriation that attempts to simulate the developmental progression reported by Piaget (1965). The model's task is to order a series of six sticks, each with a different length, so that the shortest is on the left and the longest is on the right. This is done by placing one stick in position at a time. The network is composed of two independent modules—a "which" module and a "where" module (Jacobs, Jordan & Barto, 1991). The "which" module is given the task of deciding which stick to move at a given point in the problem. The "where" module is given the task of deciding where to position each stick in terms of one of six possible spatial positions.

The results for these simulations of seriation learning match up closely with the empirical findings reported by Piaget. In stage 1, performance is close to chance. In stage 2, the network forms pairs and triplets that are correctly ordered, but the whole array is not correct. In stage 3, the whole array is ordered, but through largely trial-and-error repetition of subgroup ordering. In stage 4, seriation is performed correctly with previous analysis.

Buckingham and Shultz (1994) developed a model of the learning of the relations inherent in the physical relations expressed by these equations: $d = vt$, $v = d/t$, and $t = d/v$. These equations relate distance, velocity, and time through multiplicative relations. Wilkening (1981) found that children tend to progress through three levels of information integration in learning these relations. First, they relate each quantity only to itself. Second, they take into account the effect of the other two determining variables, but employ subtraction or addition instead of the correct division or multiplication rules. Third, they acquire the correct division or multiplication rules. Buckingham and Shultz (1994) were able to capture this three-stage developmental sequence in a neural network model. As in the other simulations reported by Shultz et al. (1995), the movement through these stages was facilitated by use of the cascade-correlation algorithm which tends to force simple solutions at early periods, but allows for the recruitment of additional resources to solve problems in more complex ways later on. In order to reach the more extreme values required by the multiplicative rule, weights have to first move through a set of values that match the additive rule. As additional units are recruited, these weights move closer to approximating a multiplicative relation.

Finally, Shultz and his colleagues have also studied the learning of resistance-potency relations. When a force with a given potency goes directly against a force with a certain resistance, the resultant force is computed by subtracting the two vectors. However, when a ramp is included in the physical system, the sum of the two vectors is computed by division, rather than subtraction. Shultz et al. (1995) were able to simulate the learning of both types of computations and showed that the subtractive relations were learned earlier than the division relations. Again, these effects seem to emerge from basic facts about the process of weight changes in neural networks.

Attachment

Van Geert (1991) developed a dynamic systems model designed to model growth curve developments in both vocabulary acquisition and the formation of attachment relations. One particularly interesting aspect of his model is the analysis he provides for the interaction between two competing developmental strategies. Van Geert shows how a variety of growth curves can arise from the competition and that the shapes of these curves depend on the internal stability of the two separate processes. In a system with optimally sensitive parenting, attachment grows steadily over time to reach a ceiling level. In a system with insensitive parenting, attachment grows weakly to reach a lower, but steady state. In a system with inconsistently sensitive parenting, the resulting attachment behavior of the child is extremely variable and unstable. These patterns of growth match up well with empirical data on the development of attachment under conditions of consistent and inconsistent parenting (Ainsworth, Blehar, Waters, & Wall, 1978; Belsky, Rovine, & Taylor, 1984).

Connectionism and Brain Development

Connectionist theory is extremely rich in terms of its implications for brain development. The first major area for which connectionism is relevant is brain development during embryogenesis. Here, connectionist models suggest that the commitment and inductance of particular neural areas to particular functions is driven by connections between areas and sensorimotor functions. The idea is that the shape of the brain emerges under the real physical constraints of the sensory and motor systems to which it is linked, rather than out of response to some abstract genetic blueprint for a set of disembodied innate ideas. To consider an example of how this works, consider the development of columns in the visual cortex (Hubel & Weisel, 1963). Miller, Keller, and Stryker (1989) have formulated network models that show how this columnar organization can arise from competitive interactions between signals from the two eyes. In general, it may be true that patterns of connectivity in the brain arise from the competition between signals arriving from different sensory systems and signals being sent to motor processes (Walsh & Cepko, 1992, 1993).

Connectionism may also help us to understand some of the mysteries of brain development during infancy and early childhood. Work on children with perinatal brain lesions (Aram & Eisele, 1992; Dennis, 1980; Feldman, 1993; Feldman, Janosky, Scher, & Wareham, 1994; Thal et al., 1991) has demonstrated the remarkable ability of the young brain to acquire normal language functioning after even the most severe early lesions. How the brain reorganizes to achieve this dynamic response is one of the great challenges facing developmental psychology and it is one in which connectionist modeling can play an important role. Recent constructivist accounts of brain development (Montague & Sejnowski, 1994; Quartz & Sejnowski, 1995) point out some possible mechanisms for changes in brain function, even after major damage. These models note that the continual refinement of patterns of connectivity is driven by local mechanisms, including dendritic growth, synaptogenesis, myelinization, and changes in membrane potential. In a constructivist model of the brain new synaptic connections are viewed as emerging through the action of nitrous oxide. When a cell fires, it broadcasts nitrous oxide to nearby cells and encourages the development of projections in the direction of the gradient of diffusion. A mechanism of this type fits in well with ideas about topological organization which we discussed earlier such as the self-organizing feature map models of Kohonen and Miikkulainen or the sparse distributed memory of Kanerva. However, a full account of reorganization after early brain damage may require more than just the local reorganization offered by these models.

The various connectionist models described in this section represent only a first step toward resolving some of the enduring issues in cognitive development. Bechtel and Abrahamsen (1991) outline the further potential of such models, including: (a) a new interpretation of the distinction between maturation and learning; (b) a computational instantiation of the distinction between accommodation and assimilation; (c) an account of context effects (in which minor task variations have large effects on preschooler's performance [Gelman, 1978]); and (d) explanations of many of the phenomena and anomalies associated with stages and transitions.

FUTURE DIRECTIONS

Having presented a description of the two principal approaches to computational modeling of cognitive development, we close with a discussion of their similarities, differences, and current inadequacies. Three themes run through this final discussion. One is that the two approaches are not as distinct as their practitioners often claim. The second is that—for all of their accomplishments—both approaches must solve some very difficult remaining problems. The third theme is that such challenges can only be met by infusing computational techniques into the training of the next generation of cognitive developmentalists.

Comparing Connectionist and Production-System Models

Although connectionist forays into cognitive development are often accompanied by the dismissal of "symbolic" approaches as unsuited to the task, we wonder whether the differences are as substantial as are sometimes claimed. Connectionist models are usually proposed as radically different from production-system architectures, and more neuronally plausible. However, one can ask where the fundamental differences lie: in the parallelism of the processing, in the distributed knowledge, or in the connectivity of that knowledge?

1. Parallelism can not be the source of the difference, because during the "match" or "recognize" phase of a production system's recognize-act cycle, the condition side of all productions are matched in parallel with all the elements in working memory.[6] In some systems, working memory is defined as the set of elements in a vast semantic memory that are above some threshold, so the match process is massively parallel and the connectivity between working memory elements and the productions is dynamic and potentially unbounded.
2. What about distributed knowledge? The extent to which knowledge is distributed or modularized in a production system depends entirely upon the grain size that elements or productions are supposed to capture. Thus, a

[6] The actual implementation of this parallel match occurs in a serial Von Neumann machine. But so too do the implementations of the learning algorithms in PDP models. This microlevel of implementation is not regarded as part of either theoretical stance.

single production might represent a very explicit and verbalizable rule; it might represent a small piece of processing for a complex, implicit piece of knowledge; or it might represent a complex pattern of cue associations much like those found in connectionist models (Ling & Marinov, 1993). Similarly, in PDP models, the individual element can represent knowledge at any grain size: from an individual neuron, to an assembly of neurons, to the word "neuron." There is nothing inherent in either formulation that specifies what this grain should be, until additional constraints are imposed on the model. Such constraints might include attempting to match production-system cycles to human reaction times, or the connectivity of connectionist models to neural connectivity.

3. Another purported difference between PDP models and production-system models is the gradualism of the former and the abruptness of the latter. But as evidenced by some of the models described earlier, one can create a production-system architecture with continuously varying strengths of productions—hence production systems can exhibit gradualism. Conversely, the higher order derivatives of different learning functions in connectionist systems can assume large values. Given the appropriate grain size on a performance window, such models would appear to be undergoing discontinuous changes (cf. Newell's 1973 classic analysis of process-structure distinctions in developmental psychology).

These many points of similarity have also been noted by advocates of the connectionist approach. Bechtel and Abrahamsen summarize some of these areas of potential overlap and rapprochement:

> Most of the modifications incorporated in the most recent symbolic models have narrowed the gap between symbolic and network models. . . . First, a large number of rules at a fine grain of analysis (microrules) can capture more of the subtleties of behavior than a smaller number of rules at a larger grain of analysis. Second, rule selection, and perhaps rule application as well, can be made to operate in parallel. Third, the ability to satisfy soft constraints can be gained by adding a strength parameter to each rule and incorporating procedures that use those values in selecting rules. Fourth, resilience to damage can be gained by building redundancy into the rule system (e.g., making multiple copies of each rule). Fifth, increased attention can be given to learning algorithms, such as the genetic algorithm (Holland, 1975; Koza, 1992), knowledge compilation and "chunking" of rules into larger units (Anderson, 1983; Newell, 1990), and ways of applying old knowledge to new problems, such as (Falkenheiner, Forbus, & Gentner, 1989).
>
> . . . There presently is no adequate research base for determining what differences in empirical adequacy might result from these differences, but the differences are likely to be small enough that empirical adequacy will not be the primary determinant of the fate of symbolic versus connectionist models. Within either tradition, if a particular inadequacy is found, design innovations that find some way around the failure are likely to be forthcoming. Personal taste, general assumptions about cognition, the sociology of science, and a variety of other factors can be expected to govern the individual choices that together will determine what approaches to cognitive modeling will gain dominance. (Bechtel & Abrahamsen, 1991, pp. 18–19)

Problems Facing Computational Models

Scalability

To date, both symbolic and subsymbolic models of cognitive development have focused on highly circumscribed domains, and within those domains, on small scale exemplars of the domain. For all of the work on connectionist models of language, no one has yet been able to construct a complete connectionist model of language acquisition. For example, developmental neural networks are often constrained to well-defined topics such as the acquisition of the English past tense (Cottrell & Plunkett, 1991), or learning German gender (MacWhinney et al., 1989). The toy model approach often reduces large problems such as question answering (St. John, 1992) or word sense disambiguation (Harris, 1994a) to small problems by using only a few dozen sentences or words in the input corpus. In fact, there is not even a reasonably complete account for smaller skill domains such as word learning or syntactic development. For all of the work on Piagetian and other types of problem solving, no one has constructed a production system or a neural net that performs the full range of tasks encountered by a normal 5-year-old child. In essence, all of the work so far has been on toy versions of larger domains.

Computational modelers argue, either explicitly or implicitly, that in principle, such models could be expanded substantially with no major theoretical modifications. But could they? Here, the plausibility of the claim varies according to the approach, with the symbolic models having the better track record. Although there are no large scale developmental production systems, there do exist several very large production systems that start with a few hundred initial "hand-coded" productions and go on to learn over 100,000 productions. Domains include both AI-type tasks and cognitive models. (See Doorenbos, 1995 for a review and evaluation of several such large-scale production systems.)

With respect to scaling up connectionist systems, there are grounds for skepticism. For example, in the language learning domain, when one attempts to add additional words or sentences to many of the connectionist language models, their performance begins to degenerate. One of the major challenges for computational models then, is a direct attack on this scalability problem.

Ad Hoc Assumptions about the Environment

Another problem facing both connectionist and production-system models is the lack of a principled, data-constrained theory of the *effective* environment in which such models operate. For many models, the "training" to which they are exposed is based on arbitrary, unprincipled, ecologically ungrounded assumptions about the environmental inputs that the child receives. Until we have better ways of measuring the actual properties of patterns in the effective environment, we cannot really claim that our models are being properly constrained by real empirical data.

Fortunately, there are two promising research avenues that may soon begin to alleviate this problem. The first avenue is the development of rich computerized databases. In the area of language development the Child Language Data Exchange System (CHILDES) database (MacWhinney, 1995) has collected transcript data from dozens of major empirical projects. These transcripts contain both the language input to the child and the child's developing conversational competence. More recently, these data are being supplemented by digitized audio and video records that give researchers access to the full richness of the original interactions. Because this database is computerized according to a standardized format, it is possible to use a wide variety of computer programs for search and analysis of patterns in both the input and the child's productions. Increasingly, simulations of language learning are being based on properties of the input as computed from the CHILDES database and similar computerized sources.

A second promising development is the growth of microgenetic studies. This research is designed to capture developmental processes as they occur by looking at fine-grained moment-to-moment changes in cognition and behavior. Kuhn (1995) has applied microgenetic techniques to the study of scientific reasoning, and Siegler and Crowley (1991) and Alibali (1993) have applied this methodology to the study of strategy development in mathematics. However, the technique can be used equally well with basic behaviors such as walking (Adolph, 1995) or reaching (Thelen & Smith, 1994). Because microgenetic methods have such a fine-grained level of analysis, they collect quantities of data that are rich enough to support interesting tests of connectionist (MacWhinney & Leinbach, 1991), symbolic (Marcus et al., 1992), and dynamic systems (van der Maas & Molenaar, 1992) approaches to cognitive development.

Hybrid Models

By now the reader has come to appreciate the degree to which connectionist models focus on low-level cognition, leaving the more complex aspects of cognitive performance to full symbolic models. There are not yet connectionist models of processes such as the learning of double-digit addition, gaining expertise in solving the Tower of Hanoi, or solving cryptarithmetic problems. Is it possible that neural networks are only appropriate as models of perception and low-level aspects of language and cognition? If so, it would make sense to graft together models that use neural networks for low-level tasks and production systems for high-level tasks.

There are reasons to believe that it would be premature to explore the construction of hybrid models of this type. Before we start building Centaurs and mermaids, we should complete our exploration of more complex, multi-componential neural network models. By linking up systems for arbitrary pattern association such as SOFM or SDM with other modules that use backpropagation or ART to extract regularities and patterns, we can increase the power of our models, while retaining the connectionist framework. When we look at the complex architecture of processing types implemented in brain structures such as the hippocampus, thalamus, and cerebellum, we realize that neuronally plausible connectionist models of tomorrow will make the simple backpropagation models of today seem primitive indeed.

Once this basic exploration of complex connectionist architectures has been completed, it may be propitious to examine the ways in which connectionist models implement algorithms developed in symbolic models such as SOAR, IBL, or ACT-R. A detailed example of close computational equivalence between a low-level symbolic model and a structured connectionist model can be found in the dialog between Ling and Marinov (1993) and MacWhinney (1993).

Why Compute?

Why should someone interested in cognitive development be concerned about computational models of the sort described in this chapter? The primary justification for focusing on such systems is the claim that self-modification is *the* central question for cognitive developmental theory. We are convinced that in order to make major theoretical advances, it will be necessary to formulate computational models at least as complex as the systems described here.

As we noted previously, early commentators on computational models often faulted them for being insufficiently attentive to the issue of self-modification. Such criticism strikes us as misplaced and ironic. While it is easy to find developmentalists who fault computational models, it is even easier to find criticisms of the entire field of developmental psychology for its inability to deal adequately with transition and change:

> I have asked some of my developmental friends where the issue stands on transitional mechanisms. Mostly, they say that developmental psychologists don't have good answers. Moreover, they haven't had the answer for so long now that they don't very often ask the question anymore—not daily, in terms of their research. (Newell, 1990, p. 462)

Is this too harsh a judgment? Perhaps we can dismiss it as based on hearsay, for Newell himself was not a developmental psychologist. But Newell's comments simply echoed an earlier assessment from one of the central figures in the field:

> . . . serious theorizing about basic mechanisms of cognitive growth has actually never been a popular pastime, . . . It is rare indeed to encounter a substantive treatment of the problem in the annual flood of articles, chapters, and books on cognitive development. The reason is not hard to find: Good theorizing about mechanisms is very, very hard to do. (Flavell, 1984, p. 189)

Even more critical is the following observation on the state of theory in perceptual development from one of the area's major contributors in recent years:

> Put simply, our models of developmental mechanisms are disappointingly vague. This observation is rather embarrassing because the aspect of perceptual developmental psychology that should set it apart from the rest of perceptual psychology is the explanation of how development occurs. and such an explanation is precisely what is lacking (Banks, 1987, p. 342)

It is difficult to deny either Newell's or Bank's assertions that we don't have good answers, or Flavell's assessment of the difficulty of the question. However, the good news is that the question is no longer being avoided: many developmentalists have been at least asking the right questions recently. In the past decade or so, we have seen Sternberg's (1984) edited volume *Mechanisms of Cognitive Development*, MacWhinney's (1987b) edited volume *Mechanisms of Language Acquisition*, and Siegler's (1989) *Annual Review* chapter devoted to transition *mechanisms*. So the question *is* being asked.

And the answers are, increasingly, coming in the form of computational models. Only a few of the chapters in the 1984 Sternberg volume specify mechanisms any more precisely than at the flow-chart level, and most of the proposed "mechanisms" are at the soft end of the information-processing spectrum. However, only five years later, Siegler (1989) in characterizing several general categories for transition mechanisms (neural mechanisms, associative competition, encoding, analogy, and strategy choice) was able to point to computationally-based exemplars for all but the neural mechanisms (Bakker & Halford, 1988; Falkenheiner et al., 1989; Holland, 1986; MacWhinney, 1987a; Rumelhart & McClelland, 1986; Siegler, 1988). The recent Simon and Halford (1995) book, consisting entirely of computational models of developmental processes, provides a clear indication of this trend toward "hardening the core" (Klahr, 1992).

The advantage of such computational models is that they force difficult questions into the foreground, where they can be neither sidetracked by the wealth of experimental results nor obscured by vague characterizations of the various "essences" of cognitive development. The relative lack of progress in theory development—noted by Banks, Flavell, and Newell—is a consequence of the fact that, until recently, most developmental psychologists have avoided moving to computationally-based theories, attempting instead to attack the profoundly difficult question of self-modification with inadequate tools. In contrast, computational models render these issues into a form sufficiently specific that it is possible to assess theoretical progress (see Mareschal & Shultz, 1996, for a cogent example).

The Future of Computational Models of Cognitive Development

That brings us to our final topic: The education of future cognitive developmentalists. As this book goes to press, the conceptual and technical skills necessary for computational modeling of developmental phenomena are taught in only a handful of graduate programs. However, we see the current situation as analogous to earlier challenges to the technical content of graduate training. When other kinds of computational technology that are now in common use—such as statistical packages—were first being applied to psychological topics, journal articles invariably included several pages of description about the technique itself. Writers of those early articles correctly assumed that their readers needed such background information before the psychological issue of interest could be addressed. Today, writers of papers using analysis of variance, or path analysis, or logistic regression simply assume that their readers have had several courses in graduate school learning the fundamentals.

Similarly, in the early years of computer simulation, the necessary resources of large "main frame" computers were limited to very few research centers, and exposure to computational modeling was inaccessible to most developmentalists. Even today, very few developmental psychologists have had any training with computational models, and only a handful of computational modelers have a primary interest in cognitive development. Nevertheless, as evidenced by the work described in this chapter, the intersection of these two areas of research is growing. Moreover, with the increasing availability of powerful workstations, the proliferation of computer networks for dissemination of computational models, the increasing number of published reports on various kinds of computationally-based cognitive architectures, the appropriate technology and support structures—such as summer workshops—are becoming widely accessible. All of these activities will increase the pool of appropriately trained developmentalists.

Even then, mastery of these new tools for computational modeling will not be easy. Nevertheless it appears to be a necessary condition for advancing our understanding of cognitive development. As Flavell and Wohlwill (1969) noted nearly thirty years ago: "Simple models will just not do for developmental psychology."

ACKNOWLEDGMENTS

We thank Robert Siegler and Chris Schunn for comments on earlier versions.

REFERENCES

Adolph, K. (1995). Psychophysical assessment of toddlers' ability to cope with slopes. *Journal of Experimental Psychology, 21,* 734–750.

Ainsworth, M. D. S., Blehar, M. C., Waters, E., & Wall, S. (1978). *Patterns of attachment: A psychological study of the Strange Situation.* Hillsdale, NJ: Erlbaum.

Albus, J. S. (1981). *Method and apparatus for implementation of the CMAC mapping algorithm.* Peterborough, NJ: McGraw-Hill.

Alibali, M. (1993). Gesture-speech mismatch and mechanisms of learning: What the hands reveal about a child's state of mind. *Cognitive Psychology, 25,* 468–523.

Alkon, D. L., Blackwell, K. T., Vogl, T. P., & Werness, S. A. (1993). Biological plausibility of artificial neural networks: Learning by non-Hebbian synapses. In M. Hassoun (Ed.), *Associative neural memories: Theory and implementation.* New York: Oxford University Press.

Anderson, J. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

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

Anderson, J., Greeno, J., Kline, P., & Neves, D. (1981). Acquisition of problem-solving skill. In J. Anderson (Ed.), *Cognitive skills and their acquisition.* Hillsdale, NJ: Erlbaum.

Anderson, J. R., Kline, P. J., & Beasley, C. M. (1978). *A general learning theory and its application to schema abstraction* (Tech Report 78-2). Pittsburgh: Carnegie Mellon University.

Aram, D. M., & Eisele, J. (1992). Plasticity and recovery of higher cortical functions following early brain injury. In F. B. J. Grafman (Ed.), *Handbook of neuropsychology: Child neuropsychology.* Amsterdam, The Netherlands: Elsevier.

Atkinson, R. D. (1975). Mnemotechnics in second-language learning. *American Psychologist, 30,* 821–828.

Atkinson, R. D., & Shiffrin, R. (1968). Human memory: A proposed system and its control processes. In K. Spence & J. Spence (Eds.), *The psychology of learning and motivation* (Vol. 2). New York: Academic Press.

Azimi-Sadjadi, M. R., Sheedvash, S., & Trujillo, F. O. (1993). Recursive dynamic node creation in multilayer neural networks. *IEEE Transactions on Neural Networks, 4,* 242–256.

Baddeley, A. D. (1986). *Working memory.* Oxford, England: Oxford University Press.

Baddeley, A D (1990) *Human memory: Theory and practice* Needham Heights, MA: Allyn & Bacon.

Bakker, P E , & Halford, G S (1988). *A basic computational theory of structure-mapping in analogy and transitive inference.* (Tech Report No 88-1): Brisbane: University of Queensland

Banks, M. (1987) Mechanisms of visual development: An example of computational models. In J. Bisanz, C. J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research.* New York: Springer-Verlag.

Baylor, G. W., & Gascon, J. (1974). An information processing theory of aspects of the development of weight seriation in children. *Cognitive Psychology, 6,* 1–40.

Bechtel, W., & Abrahamsen, A. (1991). *Connectionism and the mind: An introduction to parallel processing in networks.* Cambridge, MA: Basil Blackwell.

Behrman, M., & Bub, D. (1992). Surface dyslexia and dysgraphia: Dual routes, a single lexicon. *Cognitive Neuropsychology, 9,* 209–258.

Belsky, J., Rovine, M., & Taylor, P. (1984). The Pennsylvania Infant and Family Development Project: 3. The origins of individual differences in infant-mother attachment: Maternal and infant contributions. *Child Development, 55,* 718–728.

Besner, D., Twilley, L., McCann, R., & Seergobin, K. (1990). On the association between connectionism and data: Are a few words necessary? *Psychological Review, 97,* 432–446.

Bidell, T. R., & Fischer, K. (1994). Developmental transitions in children's early on-line planning. In M. Haith, J. Benson, R. Roberts, & B. Pennington (Eds.), *The development of future-oriented processes.* Chicago: University of Chicago Press.

Blackmore, J., & Miikkulainen, R. (1993). *Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map.* Proceedings of the IEEE International Conference on Neural Networks, San Francisco.

Block, H. D. (1962). The perception: A model for brain functioning: 1. *Review of Modern Physics, 34,* 123–135.

Bourgeois, J. P., Goldman-Rakic, P. S., & Rakic, P. (1994). Synaptogenesis in the prefrontal cortex of rhesus monkeys. *Cerebral Cortex, 4,* 78–96.

Bower, G. H. (1975). Cognitive psychology: An introduction. In E. R. Hilgard & G. H. Bower (Eds.), *Theories of learning* (pp. 25–80). Englewood Cliffs, NJ: Prentice-Hall.

Bowerman, M. (1982). Reorganizational processes in lexical and syntactic development. In E. Wanner & L. Gleitman (Eds.), *Language acquisition: The state of the art.* New York: Cambridge University Press.

Brainerd, C. (1978). The stage question in cognitive-developmental theory. *The Behavioral and Brain Sciences, 2,* 173–213.

Brown, A. L. (1982). Learning and development: The problem of compatibility, access and induction. *Human Development, 25,* 89–115.

Buckingham, D., & Shultz, T. (1994). *A connectionist model of the development of velocity, time, and distance concepts* (Proceedings of the sixteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Burgess, N., & Hitch, G. (1992). Toward a network model of the articulatory loop. *Journal of Memory and Language, 31,* 429–460.

Bybee, J. L. (1993). *Regular morphology and the lexicon.* Unpublished manuscript.

Bybee, J. L., & Slobin, D. I. (1982). Rules and schemas in the development and use of the English past. *Language, 58,* 265–289.

Campbell, D. T. (1960). Blind variation and selective retention in creative thought as in other knowledge processes. *Psychological Review, 67*(6), 380–400.

Card, S., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Erlbaum.

Carey, S. (1978). The child as word learner. In J. B. G. M. M. Halle (Ed.), *Linguistic theory and psychological reality.* Cambridge, MA: MIT Press.

Carpenter, G., Grossberg, S., Markuzon, N., Reynolds, J., & Rosen, D. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks, 3.*

Carpenter, G., Grossberg, S., & Reynolds, J. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks, 4,* 565–588.

Case, R. (1985). *Intellectual development: A systematic reinterpretation.* New York: Academic Press

Case, R. (1986). The new stage theories in intellectual development: Why we need them; what they assert. In M. Perlmutter (Ed.), *Perspectives for intellectual development* (pp. 57–91). Hillsdale, NJ: Erlbaum.

Changeux, J., & Danchin, A. (1976). Selective stabilization of developing synapses as a mechanism for the specification of neuronal networks. *Nature, 264,* 705–712.

Chauvin, Y. (1989). *Toward a connectionist model of symbolic emergence* (Proceedings of the eleventh annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Chomsky, N. (1965). *Aspects of the theory of syntax.* Cambridge, MA: MIT Press.

Clark, E. (1987). The principle of contrast: A constraint on language acquisition. In B. MacWhinney (Ed.), *Mechanisms of language acquisition* Hillsdale, NJ: Erlbaum.

Clark, E., Carpenter, K., & Deutsch, W. (1995). Reference states and reversals: Undoing actions with verbs. *Journal of Child Language, 22*, 633–652.

Cohen, D. (1983). *Piaget: Critique and assessment.* London: Croom Helm.

Coltheart, M., Curtis, B., Atkins, P., & Haller, M. (1993). Models of reading aloud: Dual-route and parallel distributed processing approaches. *Psychological Review, 100*, 589–608.

Cottrell, G. (1985). *A connectionist approach to word sense disambiguation.* Rochester, NY: University of Rochester.

Cottrell, G., & Plunkett, K. (1991). *Learning the past tense in a recurrent network: Acquiring the mapping from meaning to sounds. Proceedings of the thirteenth annual conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Craik, F. I. M., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior, 11*, 671–684.

Dechert, H., & Raupach, M. (Eds.). (1989). *Transfer in language production.* Norwood, NJ: ABLEX.

Dekaban, A. S., & Sadowsky, D. (1978). Changes in brain weights during the span of human life: Relation of brain weights to body heights and body weights. *Annals of Neurology, 4*, 345–356

Dell, G. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review, 93*, 283–321.

Dennis, M. (1980). Strokes in childhood: 1. Communication intent, expression, and comprehension after left hemisphere arteriopathy in a right-handed nine-year-old. In R. W. Reiber (Ed.), *Language development and aphasia in children.* New York: Academic Press.

Dollaghan, C., Biber, M., & Campbell, T. (1995). Lexical influences on nonword repetition. *Applied Psycholinguistics, 16*, 211–222.

Doorenbos, R. B. (1995). *Production matching for large learning systems.* Unpublished doctoral thesis, Carnegie Mellon University, Pittsburgh.

Edelman, G. (1987). *Neural Darwinism: The theory of neuronal group selection.* New York: Basic Books.

Elman, J. (1990). Finding structure in time. *Cognitive Science, 14*, 179–212.

Elman, J. (1993). Incremental learning, or the importance of starting small. *Cognition, 49*, 71–99.

Fahlman, S. E., & Lebiére, C. (1990). The cascade-correlation learning architecture. In D. Touretzky (Ed.), *Advances in neural information processing systems* (Vol. 2). Los Altos, CA: Morgan Kaufmann.

Falkenheiner, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41*, 1–63.

Fausett, D. W. (1990). Strictly local back propagation. *International Joint Conference on Neural Networks, 3*, 125–130.

Fausett, L. (1994). *Fundamentals of neural networks.* Englewood Cliffs, NJ: Prentice-Hall.

Feldman, H. M. (1993). The course of language development after early brain damage. In H. Tager-Flusberg (Ed.), *Constraints on language acquisition: Studies of atypical children.* Hillsdale, NJ: Erlbaum.

Feldman, H. M., Janosky, J. E., Scher, M. S., & Wareham, N. L. (1994). Language abilities following prematurity, periventricular brain injury, and cerebral palsy. *Journal of Communicative Disorders, 27*, 71–90.

Ferguson, C., & Farwell, C. B. (1975). Words and sounds in early language acquisition. *Language, 51*, 419–439.

Ferretti, R. P., & Butterfield, E. C. (1986). Are children's rule assessment classifications invariant across instances of problem types? *Child Development, 57*, 1419–1428.

Flavell, J. H., & Wohlwill, J. F. (1969). Formal and functional aspects of cognitive development. In D. Elkind & J. H. Flavell (Eds.), *Studies in cognitive development* (pp. 67–120). New York: Oxford University Press.

Flavell, J. J. (1984). Discussion. In R. J. Sternberg (Ed.), *Mechanisms of cognitive development* (pp. 187–210). New York: Freeman.

Flege, J., & Davidian, R. (1984). Transfer and developmental processes in adult foreign language speech production. *Applied Psycholinguistics, 5*, 323–347.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3–71.

Frean, M. (1990). The upstart algorithm: A method for constructing and training feedforward neural networks. *Neural Computation, 2*, 198–209.

Gardner-Medwin, A. R. (1976). The recall of event through the learning of associations between their parts. *Proceedings of the Yoyal Society of London B, 194*, 375–402.

Gasser, M. (1990). Connectionism and universals of second language acquisition. *Second Language Acquisition, 12*, 179–199.

Gasser, M. (1991). *Learning to recognize and produce words: Towards a connectionist model* (Proceedings of the thirteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Gasser, M. (1992). *Learning distributed representations for syllables* (Proceedings of the fourteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Gathercole, V., & Baddeley, A. (1993). *Working memory and language.* Hillsdale, NJ: Erlbaum.

Gelman, R. (1978). Cognitive development. *Annual Review of Psychology, 29*, 297–332.

Gelman, R (1982). Accessing one-to-one correspondence: Still another paper about conservation. *British Journal of Psychology, 73.* 209–220.

Gentner, D., Rattermann, M. J., Markman, A., & Kotovsky, L. (1995) Two forces in the development of relational similarity. In T. J. Simon & G. S Halford (Eds.), *Developing cognitive competence* (pp 263–314). Hillsdale, NJ: Erlbaum.

Goldberg, A. (1995). *Constructions* Chicago: University of Chicago Press.

Goodman, N. (1968). *Languages of art* Indianapolis, IN: Bobbs-Merrill.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world task performance. *Human-Computer Interaction, 8,* 237–309.

Grossberg, S. (1978). A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. *Progress in Theoretical Biology, 5,* 233–374.

Grossberg, S (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science, 11.* 23–63.

Gupta, P., & MacWhinney, B. (1992) *Integrating category acquisition with inflectional marking: A model of the German nominal system* (Proceedings of the fourteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Gupta, P., & MacWhinney, B. (1994). Is the articulatory loop articulatory or auditory? Re-examining the effects of concurrent articulation on immediate serial recall. *Journal of Memory and Language, 33,* 63–88.

Gupta, P., & MacWhinney, B. (in press). Vocabulary acquisition and verbal short-term memory: Computational and neural bases. *Brain and Language.*

Halford, G. S. (1993). *Children's understanding: The development of mental models.* Hillsdale, NJ: Erlbaum.

Halford, G S., Smith, S. B., Dickson, J. C., Mayberry, M. T., Kelly, M. E., Bain, J. D., & Stewart, J. E. M. (1995). Modeling the development of reasoning strategies: The roles of analogy, knowledge, and capacity. In T. Simon & G. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling.* Hillsdale, NJ: Erlbaum.

Hancin-Bhatt, B. (1994). Segment transfer: A consequence of a dynamic system. *Second Language Research, 10,* 241–269.

Harley, T., & MacAndrew, S. (1992). *Modelling paraphasias in normal and aphasic speech* (Proceedings of the fourteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Harrington, M. (1987). Processing transfer: Language-specific strategies as a source of interlanguage variation. *Applied Psycholinguistics, 8,* 351–378.

Harris, C. (1990). Connectionism and cognitive linguistics. *Connection Science, 2,* 7–33

Harris, C (1994a). Coarse coding and the lexicon. In C. Fuchs & B. Victorri (Eds.), *Continuity in linguistic semantics* (pp. 205–229). Amsterdam, The Netherlands: John Benjamins.

Harris, C. (1994b) Back-propagation representations for the rule-analogy continuum. In J. Barnden & K. Holyoak (Eds.), *Analogical connections* (pp 282–326) Norwood, NJ: ABLEX.

Hebb, D. (1949). *The organization of behavior.* New York: Wiley.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the theory of neural computation.* New York: Addison-Wesley.

Hetherington, P. A., & Seidenberg, M. S. (1989). *Is there "catastrophic interference" in connectionist networks?* (Proceedings of the eleventh annual conference of the Cognitive Science Society) (pp. 26–33). Hillsdale, NJ: Erlbaum.

Hinton, G., & Shallice, T. (1991) Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review, 98*(1), 74–95.

Hinton, G E., & Plaut, D. C. (1987). *Using fast weights to deblur old memories* (Proceedings of the ninth annual conference of the Cognitive Science Society) (pp. 177–186). Hillsdale, NJ: Erlbaum.

Hirose, Y., Yamashita, K., & Hijiya, S. (1991). Back-propagation algorithm which varies the number of hidden units. *Neural Networks, 4,* 61–66.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Holland, J. H. (1986). Escaping brittleness: The possibilities of general purpose machine learning algorithms applied to parallel rule-based systems. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (pp. 593–624). Los Altos, CA: Morgan-Kaufmann.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79,* 2554–2558.

Houghton, G. (1990). The problem of serial order: A neural network model of sequence learning and recall. In R. Dale, C. Mellish, & M. Zock (Eds.), *Current research in natural language generation.* London: Academic Press.

Hubel, D., & Weisel, T. (1963). Receptive fields of cells in striate cortex of very young, visually inexperienced kittens. *Journal of Neurophysiology, 26,* 994–1002.

Hunt, E. (1971). What kind of computer is man? *Cognitive Psychology, 2,* 57–98.

Hunter, I. M. L. (1968). Mental calculation. In P. C. Wason & P. N. Johnson-Laird (Eds.), *Thinking and reasoning* (pp. 341–351). Baltimore: Penguin Books.

Jacobs, R., Jordan, M., & Barto, A. (1991). Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science, 15,* 219–250.

Jerne, N. (1967). Antibodies and learning: Selection versus instruction. In G. C. Quarton, T. Melnechuk, & F. O. Schmitt (Eds.), *The neurosciences: A study program.* New York: Rockefeller University Press.

Jernigan, T. L., Archibald, S. L., Berhow, M. T., Sowell, E. R., Foster, D. S., & Hesselink, J. R. (1991). Cerebral structure on MRI: Part 1. Localization of age-related changes. *Biological Psychiatry, 29,* 55–67.

Johnson, J. (1989). Factors related to cross-language transfer and metaphor interpretation in bilingual children. *Applied Psycholinguistics, 10,* 157–177.

Jordan, M. (1986). *Serial ordering: A parallel distributed processing approach* (ICS Report 8604). La Jolla: University of California.

Juliano, C., & Tanenhaus, M. (1993). *Contingent frequency effects in syntactic ambiguity resolution* (Proceedings of the fifteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Kadirkamanathan, V., & Niranjan, M. (1993). A function estimation approach to sequential learning with neural networks. *Neural Computation, 5,* 954–975.

Kail, R. (1988). Developmental functions for speeds of cognitive processes. *Journal of Experimental Child Psychology, 45,* 339–364.

Kail, R. (1991). Processing time declines exponentially during childhood and adolescence. *Developmental Psychology, 27,* 259–268.

Kandel, E. R., & Hawkins, R. D. (1992). The biological basis of learning and individuality. *Scientific American, 266,* 40–53.

Kanerva, P. (1993). Sparse distributed memory and related models. In M. Hassoun (Ed.), *Associative neural memories: Theory and implementation.* New York: Oxford University Press.

Karmiloff-Smith, A. (1992). *Beyond modularity: A developmental perspective on cognitive science.* Cambridge, MA: MIT Press.

Kawamoto, A. (1993). Non-linear dynamics in the resolution of lexical ambiguity: A parallel distributed processing account. *Journal of Memory and Language, 32,* 474–516.

Kawamoto, A., & Zemblidge, J. (1992). Pronunciation of homographs. *Journal of Memory and Language, 31,* 349–374.

Kay, D. A., & Anglin, J. M. (1982). Overextension and underextension in the child's expressive and receptive speech. *Journal of Child Language, 9,* 83–98.

Klahr, D. (1982). Non-monotone assessment of monotone development: An information processing analysis. In S. Strauss & R. Stavy (Eds.), *U-shaped behavioral growth.* New York: Academic Press.

Klahr, D. (1992). Information processing approaches to cognitive development. In M. H. Bornstein & M. E. Lamb (Eds.), *Developmental psychology: An advanced textbook* (3rd ed.). Hillsdale, NJ: Erlbaum.

Klahr, D., & Siegler, R. S. (1978). The representation of children's knowledge. In H. W. Reese & L. P. Lipsitt (Eds.), *Advances in child development and behavior* (Vol. 12, pp. 61–116). New York: Academic Press.

Klahr, D., & Wallace, J. G. (1976). *Cognitive development: An information-processing view.* Hillsdale, NJ: Erlbaum.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43,* 59–69.

Kolers, P. A., & Smythe, W. E. (1984). Symbol manipulation: Alternatives to the computational view of mind. *Journal of Verbal Learning and Verbal Behavior, 23,* 289–314.

Köpcke, K.-M. (1994). Zur rolle von schemata bei der Pluralbildung monosyllabischer Maskulina. *Linguistische Arbeit, 319,* 81–93.

Köpcke, K.-M., & Zubin, D. (1983). Die kognitive Organisation der Genuszuweisung zu den einsilbigen Nomen der deutschen Gegenwartssprache. *Zeitschrift fur germanistische Linguistik, 11,* 166–182.

Köpcke, K.-M., & Zubin, D. (1984). Sechs Prinzipien fur die Genuszuweisung im Deutschen: Ein Beitrag zur natürlichen Klassifikation. *Linguistische Berichte, 93,* 26–50.

Kortge, C. A. (1990). *Episodic memory in connectionist networks* (Proceedings of the thirteenth annual conference of the Cognitive Science Society) (pp. 764–771). Hillsdale, NJ: Erlbaum.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection and genetics.* Cambridge, MA: Bradford Books.

Kruschke, J. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99,* 22–44.

Kuhn, D. (1995). Microgenetic study of change: What has it told us? *Psychological Science, 6,* 133–139.

Lachter, J., & Bever, T. (1988). The relation between linguistic structure and associative theories of language learning: A constructive critique of some connectionist learning models. *Cognition, 28,* 195–247.

Lamb, S. (1966). *Outline of stratificational grammar.* Washington, DC: Georgetown University Press.

Larkin, J. H. (1981). Enriching formal knowledge: A model for learning to solve textbook physics problems. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 311–334). Hillsdale, NJ: Erlbaum.

Lewis, C. (1978). *Production system models of practice effects.* University of Michigan.

Lewis, C. (1981). Skill in algebra. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 85–110). Hillsdale, NJ: Erlbaum.

Li, P., & MacWhinney, B. (1996). Cryptotype, overgeneralization, and competition: A connectionist model of the learning of English reversive prefixes. *Connection Science, 8,* 3–30.

Lightfoot, D. (1989). The child's trigger experience: Degree-0 learnability. *Behavioral and Brain Sciences, 12,* 321–275.

Ling, C., & Marinov, M. (1993). Answering the connectionist challenge. *Cognition, 49,* 267–290.

MacWhinney, B. (1978). The acquisition of morphophonology. *Monographs of the Society for Research in Child Development, 43*(Whole no. 1).

MacWhinney, B. (1982). Basic syntactic processes. In S. Kuczaj (Ed.), *Language acquisition: Vol. 1. Syntax and semantics.* Hillsdale, NJ: Erlbaum.

MacWhinney, B. (1987a). The Competition Model. In B. MacWhinney (Ed.), *Mechanisms of language acquisition.* Hillsdale, NJ: Erlbaum.

MacWhinney, B. (Ed.). (1987b). *Mechanisms of language acquisition.* Hillsdale, NJ: Erlbaum.

MacWhinney, B. (1988). Competition and teachability. In R. Schiefelbusch & M. Rice (Eds.), *The teachability of language.* New York: Cambridge University Press.

MacWhinney, B. (1989). Competition and lexical categorization. In R. Corrigan, F. Eckman, & M. Noonan (Eds.), *Linguistic categorization.* New York: Benjamins.

MacWhinney, B. (1990). Connectionism as a framework for language acquisition theory. In J. Miller (Ed.), *Progress in research on child language disorders.* Austin, TX: Pro-Ed.

MacWhinney, B. (1992). Transfer and competition in second language learning. In R. Harris (Ed.), *Cognitive processing in bilinguals.* Amsterdam, The Netherlands: Elsevier.

MacWhinney, B. (1993). Connections and symbols: Closing the gap. *Cognition, 49,* 291–296.

MacWhinney, B. (1995). *The CHILDES Project: Tools for analyzing talk* (2nd ed.). Hillsdale, NJ: Erlbaum.

MacWhinney, B. (1996). Lexical connectionism. In P. Broeder & J. M. J. Murre (Eds.), *Models of language acquisition: Inductive and deductive approaches.* Cambridge, MA: MIT Press.

MacWhinney, B., & Anderson, J. (1986). The acquisition of grammar. In I. Gopnik & M. Gopnik (Eds.), *From models to modules.* Norwood, NJ: ABLEX.

MacWhinney, B., & Bates, E. (Eds.). (1989). *The crosslinguistic study of sentence processing.* New York: Cambridge University Press.

MacWhinney, B., & Leinbach, J. (1991). Implementations are not conceptualizations: Revising the verb learning model. *Cognition, 29,* 121–157.

MacWhinney, B., Leinbach, J., Taraban, R., & McDonald, J. L. (1989). Language learning: Cues or rules? *Journal of Memory and Language, 28,* 255–277.

Maratsos, M., & Chalkley, M. (1980). The internal language of children's syntax: The ontogenesis and representation of syntactic categories. In K. Nelson (Ed.), *Children's language* (Vol. 2). New York: Gardner Press.

Marchman, V. (1992). Constraint on plasticity in a connectionist model of the English past tense. *Journal of Cognitive Neuroscience, 5,* 215–234.

Marcus, G., Ullman, M., Pinker, S., Hollander, M., Rosen, T., & Xu, F. (1992). Over regularization in language acquisition. *Monographs of the Society for Research in Child Development, 57*(4).

Mareschal, D., & Shultz, T. R. (1993). *A connectionist model of the development of seriation* (Proceedings of the fifteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Mareschal, D., & Shultz, T. R. (1996). Generative connectionist networks and constructivist cognitive development. *Cognitive Development, 11,* 571–603.

Markey, K. (1994). *The sensorimotor foundations of phonology.* Boulder, CO: University of Colorado.

Markman, E. (1989). *Categorization and naming in children: Problems of induction.* Cambridge, MA: MIT Press.

Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology, 202,* 437–470.

Marshall, J. C., & Newcombe, F. (1973). Patterns of paralexia: A psycholinguistic approach. *Journal of Psycholinguistic Research, 2,* 175–199.

Marslen-Wilson, W. D. (1987). Functional parallelism in spoken word-recognition. *Cognition, 25,* 71–102.

McClelland, J. L. (1989). Parallel distributed processing: Implications for cognition and development. In R. G. M. Morris (Ed.), *Parallel distributed processing: Implications for psychology and neurobiology.* Oxford, England: Oxford University Press.

McClelland, J. L. (1995). A connectionist perspective on knowledge and development. In T. J. Simon & G. S. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling.* Hillsdale, NJ: Erlbaum.

McClelland, J. L., & Kawamoto, A. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing.* Cambridge, MA: MIT Press.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of the basic findings. *Psychological Review, 88,* 375–402.

McClelland, J. L., & Rumelhart, D. E. (1985). Distributed memory and the representation of general and specific information. *Journal of Experimental Psychology: General. 114,* 159–188.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In G. Bower (Ed.), *The psychology of learning and motivation* (Vol. 23). New York: Academic Press.

McCorduck, P. (1979). *Machines who think.* San Francisco: Freeman.

McDonald, J. L. (1987). Assigning linguistic roles: The influence of conflicting cues. *Journal of Memory and Language, 26,* 100–107.

McDonald, J. L. (1989). The acquisition of cue-category mappings. In B. MacWhinney & E. Bates (Eds.), *The crosslinguistic study of language processing.* New York: Cambridge University Press.

Miikkulainen, R. (1990). *A distributed feature map model of the lexicon* (Proceedings of the twelfth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Miikkulainen, R. (1993). *Subsymbolic natural language processing.* Cambridge, MA: MIT Press.

Miller, K., Keller, J., & Stryker, M. (1989). Ocular dominance column development: Analysis and simulation. *Science, 245,* 605–615.

Miller, P. H. (1983). *Theories of developmental psychology.* San Francisco: Freeman.

Minsky, M., & Papert, S. (1969). *Perceptions.* Cambridge, MA: MIT Press.

Montague, P., & Sejnowski, T. (1994). The predictive brain: Temporal coincidence and temporal order in synaptic learning mechanisms. *Learning and Memory, 1,* 1–33.

Mugdan, J. (1977). *Flexionsmorphologie und Psycholinguistik.* Tübingen: Gunter Narr.

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

Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing* (pp. 463–526). New York: Academic Press.

Newell, A. (1980). Physical symbol systems. *Cognitive Science, 4,* 135–183.

Newell, A. (1981). Reasoning problem solving and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.), *Attention and performance* (Vol. 8, pp. 693–718). Hillsdale, NJ: Erlbaum.

Newell, A. (1990). *A unified theory of cognition.* Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Norman, D. A., Rumelhart, D. E., & Group, L. R. (1975). *Explorations in cognition.* San Francisco: Freeman.

Norris, D. (1994). Shortlist: A connectionist model of continuous speech recognition. *Cognition, 52,* 189–234.

Odlin, T. (1989). *Language transfer: Cross-linguistic influence in language learning.* New York: Cambridge University Press.

Oshima-Takane, Y., Goodz, E., & Derevensky, J. L. (in press). Birth order effects on early language development: Do second children learn from overheard speech? *Child Development.*

Palmer, S. E., & Kimchi, R. (1986). The information processing approach to cognition. In T. J. Knapp & L. C. Robertson (Eds.), *Approaches to cognition: Contrasts and controversies* (pp. 37–77). Hillsdale, NJ: Erlbaum.

Pascual-Leone, J. (1970). A mathematical model for the transition rule in Piaget's developmental stages. *Acta Psychologica, 32,* 301–345.

Piaget, J. (1953). *Logic and psychology.* Manchester, England: Manchester University Press.

Piaget, J. (1965). *The child's conception of number.* New York: Norton.

Piaget, J. (1975). *L'equilibration des structures cognitives* [The equilibration of cognitive structures]. Paris: Presses Universitaires de France.

Piatelli-Palmarini, M. (1989). Evolution, selection, and cognition: From "learning" to parameter setting in biology and in the study of language. *Cognition, 31,* 1–44.

Pinker, S. (1984). *Language learnability and language development.* Cambridge, MA: Harvard University Press.

Pinker, S. (1991). Rules of language. *Science, 253,* 530–535.

Pinker, S. (1994). *The language instinct.* New York: Morrow.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a Parallel Distributed Processing Model of language acquisition. *Cognition, 29,* 73–193.

Platt, J. C. (1991). A resource-allocating network for function interpolation. *Neural Computation, 3,* 213–225.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1995). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review, 103,* 56–115.

Plunkett, K., & Marchman, V. (1991). U-shaped learning and frequency effects in a multi-layered perceptron: Implications for child language acquisition. *Cognition, 38,* 43–102.

Plunkett, K., & Sinha, C. (1992). Connectionism and developmental theory. *British Journal of Development Psychology, 10,* 209–254.

Posner, M., & Keele, S. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology, 77,* 353–363

Posner, M., & Keele, S (1970). Retention of abstract ideas. *Journal of Experimental Psychology, 83,* 304–308.

Prasada, S., & Pinker, S. (1993). Generalization of regular and irregular morphological patterns. *Language and Cognitive Processes, 8,* 1–56.

Quartz, S. R., & Sejnowski, T. J. (in press). The neural basis of cognitive development: A constructivist manifesto. *Behavioral and Brain Sciences*

Rabinowitz, R. M., Grant, M. J., & Dingley, H. L. (1987). Computer simulation, cognition, and development: An introduction. In J. Bisanz, C. J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research* (pp. 263–301). New York: Springer-Verlag.

Read, W., Nenov, V. I., & Halgren, E. (1995). Inhibition-controlled retrieval by an autoassociative model of hippocampal area CA3. *Hippocampus, 5,* 79–91.

Rosenblatt, F. (1959). Two theorems of statistical separability in the perception. In M. Selfridge (Ed.), *Mechanisation of thought processes: Proceedings of a symposium held at the National Physical Laboratory.* London: HM Stationery Office.

Rosenbloom, P. S., & Newell, A. (1982). *Learning by chunking: Summary of a task and a model.* Paper presented at the second national conference on Artificial Intelligence, Los Altos, CA.

Rosenbloom, P. S., & Newell, A. (1987). Learning by chunking: A production system model of practice. In D. Klahr, P. Langley, & R. Neches (Eds.), *Production system models of learning and development* (pp. 221–286). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tense of English verbs. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition.* Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processes? In B. MacWhinney (Ed.), *Mechanisms of language acquisition.* Hillsdale, NJ: Erlbaum.

St. John, M. (1992). The story gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science, 16,* 271–306.

Sasaki, Y. (1994). Paths of processing strategy transfers in learning Japanese and English as foreign languages. *Studies in Second Language Acquisition, 16,* 43–72.

Schmajuk, N., & DiCarlo, J. (1992). Stimulus configuration, classical conditioning, and hippocampal function. *Psychological Review, 99,* 268–305.

Schyns, P. (1991). A modular neural network model of concept acquisition. *Cognitive Science, 15,* 461–508.

Seidenberg, M., & McClelland, J. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review, 96,* 523–568.

Sejnowski, T. J., & Rosenberg, C. R. (1988). NETtalk: A parallel network that learns to read aloud. In J. A. Anderson & E. Rosenfeld (Eds.), *Neurocomputing: Foundations of research.* Cambridge, MA: MIT Press.

Shultz, T., Buckingham, D., & Oshima-Takane, Y. (1994). A connectionist model of the learning of personal pronouns in English. In S. J. Hanson, T. Petsche, M. Kearns, & R. L. Rivest (Eds.), *Computational learning theory and natural learning systems: Vol. 2. Intersection between theory and experiment.* Cambridge, MA: MIT Press.

Shultz, T., Schmidt, W. C., Buckingham, D., & Mareschal, D. (1995). Modeling cognitive development with a generative connectionist algorithm. In T. J. Simon & G. S. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling.* Hillsdale, NJ: Erlbaum.

Siegler, R. (Ed.). (1978). *Children's thinking: What develops?* Hillsdale, NJ: Erlbaum.

Siegler, R. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experiment Psychology: General, 117,* 258–275.

Siegler, R. (1989). Mechanisms of cognitive development. *Annual Review of Psychology, 40,* 353–379.

Siegler, R., & Crowley, K. (1991). The microgenetic method: A direct means for studying cognitive development. *American Psychologist, 46,* 606–620.

Siegler, R., & Shrager, J. (1984). Strategy choices in addition and subtraction: How do children know what to do? In C. Sophian (Ed.), *Origins of cognitive skills.* Hillsdale, NJ: Erlbaum.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology, 8,* 481–520.

Siegler, R. S., & Shipley, C. (1995). Variation, selection, and cognitive change. In G. Halford & T. Simon (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 31–76). Hillsdale, NJ: Erlbaum.

Simon, D. (1976). Spelling, a task analysis. *Instructional Science, 5,* 277–302.

Simon, D., & Simon, H. (1973). Alternative uses of phonemic information in spelling. *Review of Educational Research, 43,* 115–137.

Simon, D., & Simon, H. (1978). Individual differences in solving physics problems. In R. Siegler (Ed.), *Children's thinking: What develops?* (pp. 325–348). Hillsdale, NJ: Erlbaum.

Simon, H. A. (1962). An information processing theory of intellectual development. *Monographs of the Society for Research in Child Development, 27.*

Simon, T., & Klahr, D. (1995). A theory of children's learning about number conservation. In T. Simon & G. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling*. Hillsdale, NJ: Erlbaum.

Simon, T., & Halford, G. (Eds.). (1995). *Developing cognitive competence: New approaches to process modeling*. Hillsdale, NJ: Erlbaum.

Stemberger, J. (1985). *The lexicon in a model of language production*. New York: Garland.

Sternberg, R. J. (Ed.). (1984). *Mechanisms of cognitive development*. New York: Freeman.

Thal, D. J., Marchman, V. A., Stiles, J., Aram, D., Trauner, D., Nass, R., & Bates, E. (1991). Early lexical development in children with focal brain injury. *Brain and Language, 40*(4), 491–527.

Thelen, E., & Smith, L. (1994). *A dynamic systems approach to the development of cognition and action*. Cambridge, MA: MIT Press.

Touretzky, D. (1990). BoltzCONS: Dynamic symbol structures in a connectionist network. *Artificial Intelligence, 46*, 5–46.

Trueswell, J. C., & Tanenhaus, M. K. (1991). Tense, temporal context, and syntactic ambiguity resolution. *Language and Cognitive Processes, 6*, 303–338.

Trueswell, J. C., & Tanenhaus, M. K. (1992). *Consulting temporal context during sentence comprehension: Evidence from the monitoring of eye movements in reading* (Proceedings of the fourteenth annual conference of the Cognitive Science Society). Hillsdale, NJ: Erlbaum.

Twain, M. (1935). *The awful German language. The family Mark Twain*. New York: Harper & Brothers.

van der Maas, H., & Molenaar, P. (1992). Stagewise cognitive development: An application of catastrophe theory. *Psychological Review, 99*, 395–417.

van Geert, P. (1991). A dynamic systems model of cognitive and language growth. *Psychological Review, 98*, 3–53.

von Neumann, J. (1956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. Shannon & J. McCarthy (Eds.), *Automata studies*. Princeton, NJ: Princeton University Press.

Vygotsky, L. (1962). *Thought and language*. Cambridge, MA: MIT Press.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. (1988). Phoneme recognition using time-delay neural networks. *IEE Transactions on Acoustics, Speech and Signal Processing, 37*.

Wallace, J. G., Klahr, D., & Bluff, K. (1987). A self-modifying production system for conservation acquisition. In D. Klahr, P. Langley, & R. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.

Walsh, C., & Cepko, C. L. (1992). Clonally related cortical cells show several migration patterns. *Science, 255*, 434–440

Walsh, C., & Cepko, C. L. (1993). Widespread dispersion of neuronal clones across functional regions of the cerebral cortex. *Science, 362*, 632–635.

Waterman, D. (1975). *Adaptive production systems*. Proceedings of the Fourth International Joint Conference on Artificial Intelligence, 296–303. Tbilisi, USSR.

Watrous, R. L., Shastri, L., & Waibel, A. H. (1987). Learned phonetic discrimination using connectionist networks. In J. Laver & M. A. Jack (Eds.), *Proceedings of the European conference on Speech Technology*. Edinburgh: CEP Consultants.

Waugh, E., & Norman, D. (1965). Primary memory. *Psychological Review, 72*, 89–104.

Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Unpublished Ph.D. thesis, Harvard University, Cambridge, MA.

Whorf, B. (1938). Some verbal categories of Hopi. *Language, 14*, 275–286.

Whorf, B. (1941). The relation of habitual thought and behavior to language. In L. Spier (Ed.), *Language, culture, and personality: Essays in memory of Edward Sapir*. Ogden: University of Utah Press.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *IRE WESCON Convention Record* (part 4), 96–104.

Wilkening, F. (1981). Integrating velocity, time, and distance information: A developmental study. *Cognitive Psychology, 13*, 231–247.

Wilkening, F., & Anderson, N. H. (1982). Representation and diagnosis of knowledge structures in developmental psychology. In N. H. Anderson (Ed.), *Contributions to integration theory: Vol. 3. Developmental*. Hillsdale, NJ: Erlbaum.

Wynne-Jones, M. (1993). Node splitting: A constructive algorithm for feed-forward neural networks. *Neural Computing and Applications, 1*, 17–22.

Young, R. M. (1976). *Seriation by children: An artificial intelligence analysis of a Piagetian task*. Basel, Switzerland: Birkhauser.

Zubin, D. A., & Köpcke, K. M. (1981). Gender: A less than arbitrary grammatical category. In C. M. R. Hendrick & M. Miller (Eds.), *Papers from the seventeenth regional meeting*. Chicago: Chicago Linguistic Society.

Zubin, D. A., & Köpcke, K. M. (1986). Gender and folk taxonomy: The indexical relation between grammatical and lexical categorization. In C. Craig (Ed.), *Noun classes and categorization*. Amsterdam, The Netherlands: John Benjamins.