

From: E.K. Scholnick, K. Nelson, S.A. Gelman, & P.H. Miller (Eds.). Conceptual development: Piaget's legacy. Mahwah, NJ: Lawrence Erlbaum Associates.

Chapter **6**

## The Conceptual Habitat: In What Kind of System Can Concepts Develop?

David Klahr

*Carnegie Mellon University, Pittsburgh, PA*

Some theories of conceptual development focus on the content of domain-specific conceptual acquisitions (e.g., Carey, 1985), while other theories emphasize domain-general processes that support the various acquisitions (e.g., Halford, 1993). A few offer balanced accounts of content and process, but in limited domains such as arithmetic (e.g., Siegler, 1996; Siegler & Shipley, 1995). In this chapter, I focus on a third aspect of conceptual development by addressing the following question: In what kind of system can conceptual development occur? This question differs from questions about the content or process of conceptual development because it addresses the nature of the *underlying system* that represents content and executes processes.

This difference can best be understood in terms of the following analogy. Consider the situation you find yourself in when you purchase a new piece of software. You know that the compact disk (CD) contains all the data and programs required to function properly, but something else is necessary before the programs and data can become operational. They can work only if you have a computer with both a minimal hardware capacity (i.e., disk space, random-access memory capacity, monitor specifications) and a minimal level of operating system. If any of these constraints is not met, you cannot use the CD. Now imagine that your CD contains a "universal conceptual development kit." (See Fig. 6.1.) Do you have a system that can handle it? Analogously, given any particular theoretical statement about the mechanisms of conceptual development, we can ask: What kind of

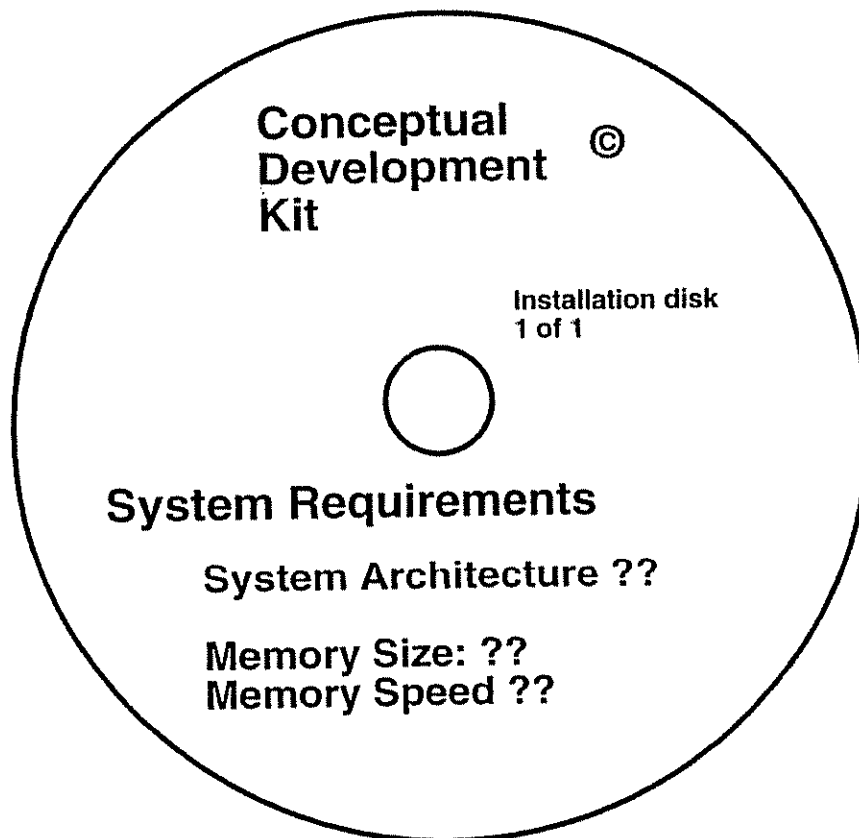


FIG. 6.1. An imaginary compact disk containing a conceptual development system.

mental architecture is necessary to support the concepts and processes proposed by that theory?

Just as ecologists find it necessary to characterize the ecological niche of their focal species in order to fully understand their evolution and survival, psychologists need to ask about the nature of the system in which conceptual development takes place. In other words, they need to ask: "What is the conceptual habitat?"

In this chapter, I suggest that self-modifying computational models provide a means of answering this fundamental question. They do so not only by providing detailed accounts of a variety of phenomena associated with conceptual development, but also, and more important, by providing theories of the human cognitive architecture.

Newell (1990) defined a cognitive architecture as "the fixed (or slowly varying) structure that forms the framework for the immediate processes of cognitive performance and learning" (p. 12). What is the nature of this cognitive architecture? How is it organized? What are the computational principles and constraints under which it operates? Such questions define

the research frontiers for those formulating computational models of thought.

Two relatively distinct approaches to computational modeling of developmental phenomena have emerged along these frontiers: production systems and connectionist systems. Production systems have tended to focus on symbolically based, rule-oriented, higher cognitive processes, whereas connectionist systems have focused on the subsymbolic (or nonsymbolic), neurally analogous, microstructure of cognition (see T. J. Simon & Halford, 1995; Klahr & MacWhinney, 1997, for extensive descriptions and comparisons of these approaches).

### **PIAGET'S ATTEMPTS TO CHARACTERIZE THE SYSTEM**

Before describing these computational approaches, I start with a bit of history. The effort to characterize the system in which concepts develop is not a new endeavor: Piaget's legacy is his lifelong inquiry about the dynamic system in which conceptual development occurs. Piaget characterized this system in terms of the formalisms available to him at the time. From logic and mathematics, he constructed a representational system. From biology, he borrowed the notion of assimilation and accommodation (cf. Case, 1997). However, Piaget's initial characterization of these processes was at a highly abstract level, and no one has yet figured out how to translate his ideas into an unambiguous operational system.

Even Piaget was dissatisfied with his early formulation of the equilibration process, and he continually reconceptualized and refined it. Thus, as late as 1975, he was using representations like the one in Fig. 6.2 to describe assimilation and accommodation. Although the notation gives the appearance of a more precisely conceptualized account of equilibration, the accompanying text makes it clear that the mechanisms it depicts remain obscure. "Before sufficiently precise models are achieved, therefore, one witnesses a succession of states indicating progressive equilibration. The initial states of this progression achieve unstable forms of equilibrium only because of lacunae, because of perturbations, and above all, because of real or potential contradictions" (Piaget, 1985/1975, p. 47).

### **COMPUTATIONAL MODELS OF COGNITIVE DEVELOPMENT**

About 30 years ago, at the same time that U.S. psychologists began to wrestle with Piaget's ideas, there emerged in this country what has been called "the cognitive revolution" and with it the information-processing

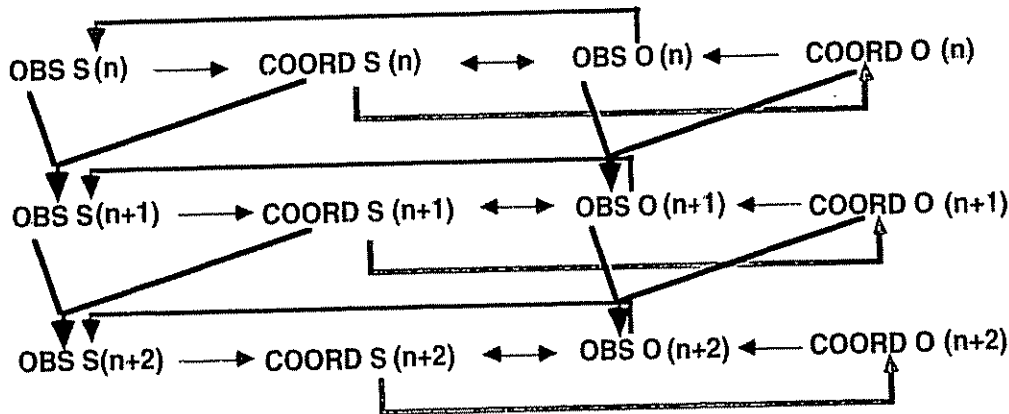


FIG. 6.2. A depiction of part of Piaget's model of assimilation and accommodation. OBS S—"observables relative to the subject's action"; OBS O—"observables relative to objects"; COORD S—"inferential coordinations of the subject's actions or operations"; COORD O—"inferential coordinations among objects." From *The Equilibration of Cognitive Structures* (p. 44), by J. Piaget, 1985, Chicago: University of Chicago Press. Copyright 1985 by the University of Chicago Press. Reprinted with permission.

approach to cognitive development (R. Brown, 1970; Klahr, 1992; H. A. Simon, 1962). Fig. 6.3 presents a concise depiction of its essential ideas. The most important of these is that cognitive theories can be stated as computer programs. This idea is not only fundamentally important but also widely misunderstood. Many people have argued that computational modelers equate the human mind to a digital computer (A. L. Brown, 1982; Miller, 1983). As Mark Antony said of Julius Caesar's ambition: "If it were so, it were a grievous fault." But it is *not* so.

Perhaps the misconception can be corrected by considering an example from another field in which computational models play a central role. Meteorologists who run computer simulations of hurricanes do not believe that the atmosphere works like a computer or that their models generate fog, rain, snow, or sunshine. They *do* believe that their characterizations of the atmosphere are so complex that only a computer can draw out their implications. It is the equations, the models, that are supposed to work like the atmosphere, not the computer on which the models run. They are also aware that their current models are only that: models. As such, they fail to capture many complexities, subtleties, and local anomalies of meteorological processes. So too for computational models of conceptual development: Such models assume neither that the underlying silicon bears any relation to neural tissue nor that any single model captures all of cognition.

Another idea, and the one that is important for developmentalists, is that children's knowledge at different states or levels can be described by different computational models. A third idea follows from the first two: If different states of cognitive development can be accounted for by compu-

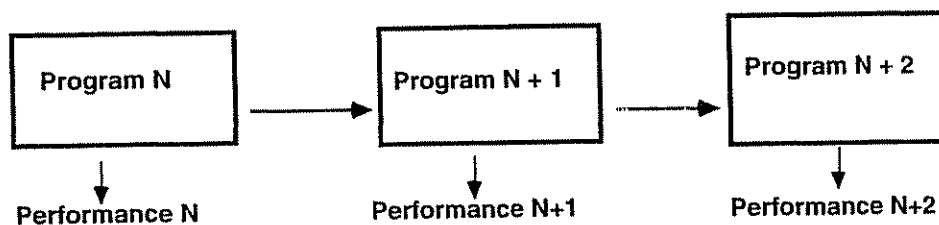
tational models (i.e., by performance models), then so too can the developmental process that produced those states (i.e., adaptation models). Such programs would have the capacity to alter and extend their own processes and structures. That is, they would be self-modifying computational models, and the model building enterprise would have two steps. First, build the sequence of state models and then build the transition model.

The earliest computational models of developmental phenomena addressed states but not transitions (Baylor & Gascon, 1974; Klahr & Wallace, 1976). The plan was that the adaptive transition models could come later, after the performance models for successive states were developed and evaluated. This two-step approach has gradually given way to models in which performance and adaptation occur simultaneously (indicated by 3' in Fig. 6.3).

It is difficult to achieve an appropriate balance between performance and adaptation. The two primary approaches to computational modeling mentioned earlier—production systems and connectionist systems—have tended to emphasize different aspects of this delicate balance: Production systems tend to emphasize performance over adaptation, whereas connectionist systems tend to emphasize adaptation over performance. Descriptions and examples of both approaches are provided later in this chapter, and it will become evident that there are important distinctions as well as some fundamental commonalities.

### 1. Cognitive theory can be stated as a computer program

(But: mind is **NOT** a computer!)



### 2. Distinct program for each knowledge level

### 3. Transition program to modify from one level to the next

### 3'. Performance and adaptation intermingled

FIG. 6.3 Basic assumptions in computation models of cognitive development.

One of the most important commonalities is the feature that distinguishes computational models from all other types of theoretical statements: *They independently execute the mental processes that they represent.* That is, rather than leaving it to readers to interpret a verbal description or a diagram of such processes as searching a problem space, redescribing a representation, or coordinating an inference, computational models actually *do* the searching, redescribing, and coordinating. This similarity, in my mind, outweighs all the real and apparent differences between symbolic and subsymbolic computational models. In fact, as I suggest later, the distinctions between the two approaches are diminishing as both devote more effort to addressing developmental issues. In the next two sections, I briefly describe each approach.

## PRODUCTION SYSTEMS

The important properties of production system architectures are listed in Table 6.1. This list describes only *current* properties, which will certainly change as we learn more about how to build adaptive production systems that capture important developmental phenomena. I review the basics of production systems and then focus on some interesting issues in the field.

### Declarative (“Working”) Memory

A production system consists of two primary structures: declarative memory and production memory. Declarative memory is used to represent objects, features, and goals. It is usually called working memory, but it is more accurate to call it declarative memory. It contains both long-term knowledge and aspects of the immediate situation such as goals and subgoals. An important design feature of different production-system architectures is the way that they resolve several related questions about the dynamics and complexity of declarative memory elements. How permanent are the declarative memory elements? Are they erased after the task is complete, or do they remain indefinitely? The basic problem is that the more infor-

TABLE 6.1  
Current Properties of Production Systems

---

Declarative memory (“working memory”) represents objects, features, and goals.
Procedural knowledge is stored as if-then rules (“productions”).
Executing (“firing,” “satisfying”) a production is the fundamental unit of thought.
Adaptation takes place through the acquisition and modification of productions.
The results of computation are stored in a (temporary?) declarative memory.
Knowledge is (mostly) modular.

---

mation there is in declarative memory, the more likely it is that many productions are satisfied simultaneously. This situation complicates the conflict resolution process.

These questions have been answered in several different ways by production system designers. At one extreme are systems in which items *do* stay around forever. At the other extreme are systems in which items are deleted once the system moves on to the next task. Intermediate between these two extremes are systems in which the elements vary in activation (which in turn determines how available or easily retrieved they are). The activation increases each time the represented facts or items are encountered and decays with time after each encounter.

### Production Memory

The second basic structure consists of a set of if-then rules or *productions* that represent skills or procedures for interacting with the world. In these productions, the *if* side is called the *condition* and the *then* side is called the *action*. The condition side of a production is a list of entities that must appear in declarative memory. When the conditions of a production are true of the current state of declarative memory, then the production is said to *fire* or *match* (or to *be satisfied*). The action side of a production can refer to either behavioral actions or new declarative memory elements representing a new piece of knowledge or a new goal. Because all productions are matched in parallel, these systems have the power to be reactive to changes in the environment and to consider large numbers of responses simultaneously.

How does the system decide what to do when more than one production's conditions are satisfied? The process by which a production system chooses among satisfied productions is called *conflict resolution*. These decisions are viewed as an integral part of the cognitive architecture. A number of conflict resolution schemes have been used over the past 25 years. These schemes include:

1. *Recency*: favoring productions whose conditions refer to declarative memory elements that have been most recently added or changed.
2. *Specificity*: favoring productions with many conditions (i.e., more specific productions) over productions with few conditions.
3. *Importance*: setting a rank ordering among productions according to a predetermined scheme.
4. *Frequency*: giving preference to productions that have been used most often and most successfully. In this way, production order can be adapted to different experience. (I return to this issue later when I discuss learning in production systems.)

A more radical approach to deciding which production to fire is to just "do it," to fire all the satisfied productions. In one such scheme, the productions make suggestions only about what to do (applying knowledge from past experiences), and then another conflict resolution scheme must decide among these suggestions (Newell, 1990). In another scheme, all productions fire, but they must compete for a limited pool of activation resources (Just & Carpenter, 1992). The full implications of these different schemes are still being determined. This problem is one of the research frontiers in the production system world.

### Self-Modification in Production Systems

How can a production system adapt, learn, and develop? At present, there are two primary mechanisms for self-modification: One set of mechanisms creates new productions, and the other modifies or tunes existing productions.

*Creating New Productions.* One way in which new productions can be created is via *compilation*, in which a new production is produced; this production does, in one step, the action of several productions. An important variant of compilation is the chunking algorithm used in the Soar production system (Newell, 1990). The *chunking* algorithm determines which pieces of declarative knowledge were used by a recently successful sequence of productions and then creates a new production that looks for these declarative memory elements and directly produces the desired conclusion without going through the intermediate steps. This mechanism is used by T. J. Simon and Klahr (1995) to account for children's learning in one of Rochel Gelman's conservation training studies (Gelman, 1982).

Another mechanism is *analogy*. This mechanism, used in Anderson's (1993) ACT-R production system, converts examples in declarative memory into productions. When no production works in the current context, the system tries to make an analogy between the current goal and the corresponding goal in an example and creates productions that achieve the current goal by using steps analogous to the ones used to achieve the source goal.

*Modifying Existing Productions.* Other production-learning mechanisms create productions by *combining* or *mutating* existing productions. For example, adding or deleting conditions to a production thereby makes it more or less situation specific. Classifiers are a special case of this process (Holland, 1975). They consist of a set of rules for classifying instances into different categories. New rules are created by randomly mutating some



conditions of existing rules. The rules that do a good job of classifying the instances are kept, whereas new rules that do a poor job are discarded.

In many domains, performance improves gradually, not abruptly. How might a production system achieve this gradualism if learning new productions creates discrete jumps in performance? One way to do this is to formulate productions at a very fine-grained level of detail such that many productions are required to produce each external action. In such a scheme, the addition of each production produces only a minor improvement in performance. Another solution is to associate with each production parameters that cause productions to perform slowly, suboptimally, or infrequently when they are first created and then gradually to become faster, more efficient, or more frequent. Productions can be strengthened according to their record of successful and unsuccessful use. Production strength can then determine the likelihood that the production is selected during the conflict resolution phase.

*Other Self-Modification Mechanisms?* One of the fundamental research questions in this area is just how many of the major phenomena of cognitive development can be explained by the self-modification processes described thus far. For example, it is not yet clear whether basic production modification processes 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, 1978; D. P. Simon & Simon, 1978). Such reorganization may involve much more than refinements in the productions governing *when* suboperations are performed. These refinements could be produced by generalization and discrimination mechanisms, but 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 suggested, but they remain to be implemented in computational models. For example, Wallace, Klahr, and Bluff (1987) proposed a production system architecture that included a hierarchically organized set of nodes, each of which is a semiautonomous production system, communicating via a shared working memory. Each of these nodes can be simultaneously activated. The basic developmental process involves the construction of new nodes by processing a representation of episodic sequences for the system's 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 on them. Spensley

(1995) has proposed an interesting integration and extension of both Wallace et al.'s and Karmiloff-Smith's proposals.

Such soft-core notions present challenges to the hard-core approach<sup>1</sup> described in this chapter: Either implement these ideas or show that they are theoretically unnecessary or create a computational alternative that accomplishes the same thing.

### Knowledge Is (Mostly) Modular

How does knowledge interact, and how does learning generalize? Production systems provide strong answers to these fundamental questions: Learning occurs at the unit of productions; transfer from one situation to another occurs to the extent that the same productions are applicable in both situations.

Because of their modularity, production systems scale up well to complex tasks. That is, production systems function well not only on small, simple tasks but also in realistic environments involving many subtasks and tens of thousands of knowledge elements. This modularity is not perfect, and some productions may not be entirely independent of all other productions. This situation can be particularly troublesome in *adaptive* production systems because new productions can interfere with the previously smooth functioning of an earlier series of productions. This problem is one of the most difficult aspects of building production system models.

## CONNECTIONIST SYSTEMS—A BRIEF OVERVIEW

All 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.<sup>2</sup> Connectionist systems use neither symbols nor rules to represent knowledge. The only sense in which they embody a cognitive architecture is their strong commitment to distributed knowledge and a loose commitment to the notion that the models are connected somewhat analogous to the way that 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

---

<sup>1</sup>See Klahr (1992) for a discussion of the distinction between hard-core and soft-core information-processing approaches in developmental psychology.

<sup>2</sup>This section is adapted from Klahr and MacWhinney (1997).

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) succinctly characterized 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. (p. 158)

Because connectionist systems are inherently learning systems, the two-step approach to modeling conceptual development described earlier (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 that 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 (e.g., Rumelhart & McClelland, 1986), but more recent models—some of which I describe next—have begun to examine conceptual development and problem solving.

### **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. Such networks 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 Fig. 6.4, the units are labeled with letters such as  $x_1$ .

2. *Connections.* Units or pools of units are connected by a set of pathways variously called connections, links, pathways, or arcs. In most models, these connections are unidirectional and go 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 Fig. 6.4, the connection between units  $x_1$  and  $y_1$  is marked with a thick line.

3. *Patterns of connectivity.* Units are typically grouped into pools or layers. Connections can operate in or between layers. In some models (such as the one shown in Fig. 6.4), there are no in-layer connections; in others, all units in a given layer are interconnected. Units or layers can be further divided into three classes:

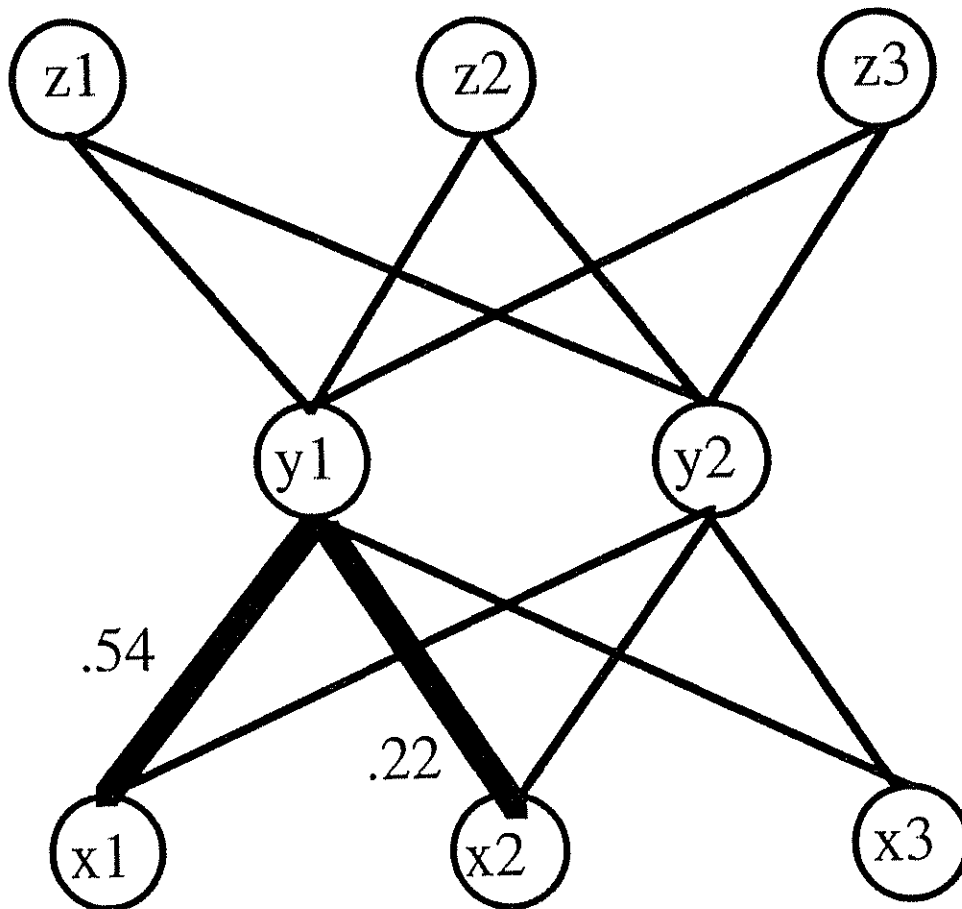


FIG. 6.4. The basic components of a connectionist model (see text for explanation).

*Input units*, which represent signals from earlier networks. These units are marked x in Fig. 6.4.

*Output units*, which represent the choices or decisions made by the network. These units are marked z in Fig. 6.4.

*Hidden units*, which represent additional units juxtaposed between input and output for the purposes of computing more complex, nonlinear relations. These units are marked y in Fig. 6.4.

4. *Weights*. Each connection has a 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 x1 and y1 is given as .54 in Fig. 6.4.

5. *Net inputs*. The total amount of input from a sending unit to a receiving unit is determined by multiplying the weights on each connection to the receiving unit by the activation of the sending unit. This net input to the receiving unit is the sum of all such inputs from sending units. In Fig. 6.4, the net input to y1 is .76, if we assume that the activations of x1 and x2 are both 1 and the x1y1 weight is .54 and the x2y1 weight is .22.

6. *Activation functions*. Each unit has a level of activation. These activation levels can vary continuously between 0 and 1. 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 in which 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.

The most common approach—called *back propagation*—is to present the network with an input pattern and to compare the output pattern it produces with the one that is desired (i.e., the thing to be learned). The system then computes the difference between these two and adjusts the weights so as to approach the desired pattern in an optimal way. The basic idea is to adjust each parameter in the network in proportion to the effect that the adjustment has on the overall fit to the desired output. Once the adjustments are made, another comparison is done, and the system reiterates this process for many cycles.

Another approach—called *cascade correlation*—has been used by Shultz, Schmidt, Buckingham, and Mareschal (1995) to model several developmental domains, including causal reasoning, seriation, integration of distance, time, and velocity, and personal pronouns. Cascade correlation mod-

els start with a network that has no hidden units. Such units are added—as part of the training–learning process—when the system decides that its rate of learning has reached a plateau.

All connectionist networks share this common language of units, connections, weights, and learning rules, but models differ markedly both in their detailed patterns of connectivity and in the specific rules used for activation and learning.<sup>3</sup>

## **TWO COMPUTATIONAL APPROACHES TO THE SAME DOMAIN: THE BALANCE SCALE**

In general, the domains in which production system models and connectionist models have been proposed have been nonoverlapping. Production systems have been used mainly to model higher order problem-solving domains, whereas connectionist models have tended to focus on perceptual and language development. In one domain, familiar to all developmentalists, however, both types of models have been formulated: Piaget's balance scale prediction task.

### **Production Systems for the Balance Scale**

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.

The basic physical concept that underlies the operation of the balance scale is torque: The scale rotates in the direction of the greater of the two torques acting on its arms. Because the pegs are at equal intervals from the fulcrum and the weights are all equal, a simple torque calculation is possible. It is the sum of the products of the number of weights on a peg times the ordinal position of the peg from the fulcrum. This calculation is done for each side, and the side with the greater sum of products is the side that goes down. (If they are equal, the scale balances.)

Siegler (1976) demonstrated that children's different levels of knowledge about this task can be represented in the form of a sequence of four increasingly mature rules or models. A child using Model I considers only

---

<sup>3</sup>For excellent, readable introductions to the theory and practice of neural network modeling, readers may wish to consult Bechtel and Abrahamsen (1991) or Fausett (1994). For a mathematically advanced treatment, see Hertz, Krogh, and Palmer (1991).

the number of weights on each side: If they are the same, the child predicts balance; otherwise he or she predicts that the side with the greater weight will go down. For a child using Model II, 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 goes 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 same side with respect to their inequality, then that side is predicted to go down. In a situation in which one side has greater weight and the other has greater distance, the child, although recognizing the conflict, does not have a consistent way to resolve it but simply muddles through by making a random prediction.

A child using Model IV represents mature knowledge of the task: Because it includes the sum-of-products calculation, children using it always make the correct prediction, but if they can base their prediction on simpler tests, they 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 solve balance scale problems consistently.

Siegler represented these different levels of knowledge in the form of binary decision trees that could make clear predictions about the responses made by a child using one of these rules for any specific configuration of weights. Such decision trees are silent on the dynamics of the decision process, however, and they do not make a clear distinction between encoding processes and decision processes. By recasting the rules as production systems, Klahr and Siegler (1978) were able to make a more precise characterization of what develops than was afforded by the decision-tree representation.

Their production system is listed in Table 6.2. For example, Model II in Table 6.2 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 6.2 contains a representation for any finer grain knowledge, such as the actual amount of weight or distance or the means used to encode that information. There is no 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.

On any recognize-act cycle, only one of these productions fires, depending on the type of knowledge that the encoding processes have placed in working memory. If the weights are unequal, then P2 fires; if the weights

TABLE 6.2  
Production System Representations for Balance Scale Models I-IV

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")]
<i>Transitional Requirements</i>	
<i>Transitions</i>	<i>Production Modifications</i>
I → II	Add P3
II → III	Add P4, P5
III → IV	Modify P4; add P6, P7
	<i>New Operators</i>
I → II	Add distance encoding and comparison.
II → III	None.
III → IV	Add torque computation and comparison.

*Note* D = Distance; W = Weight

are equal and the distances are not, then both P1 and P3 are satisfied, and this conflict must 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. Finally, if both weights and distances are equal, then only P1 is satisfied, and it fires. The task facing a transition model is indicated at the bottom of Table 6.2. 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 I have compared the models at the level of productions, but productions need information provided by the operators that encode the external configuration. 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. It requires an encoding process that either directly encodes relative weight or encodes an absolute amount of each and then inputs these



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, and 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 does any of the preceding models. To determine relative torque, it must first determine the absolute torque on each side of the scale, and this calculation requires exact numerical representation of weight and distance. In addition, the torque computation requires access to the necessary arithmetic production systems to actually do the sum of products calculations.

Although I have compared the four models at two distinct levels—productions and operators—the levels are not that easily separated. Missing from these models is a set of productions that indicates the interdependence: productions that explicitly determine which encoding the system makes. In these models, there are almost no productions of the form: (want to compare weights)  $\rightarrow$  (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 fires 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 previously, produces 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, but it is instructive to consider the way in which such detailed models could characterize how much more than balance scale knowledge, as such, is required by a child

performing this task. For example, one of Klahr and Siegler's subjects tended to encode both weight and distances as either big or small. Their model for that subject dealt with the way in which the child maintained declarative memory elements representing 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 (for big weights or distances), 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 encoded knowledge must be available at any one moment and hence about the dynamics of declarative memory mentioned earlier. 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 work by Halford and his colleagues (Halford, 1993; Halford et al., 1995).

The balance scale production systems exemplify the sequence-of-stages approach used in the early days 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 children's responses to these tasks. Although, as noted earlier, adaptive production systems exist in other domains, as yet there is no such adaptive production system for the balance scale domain. This area is one of the few involved in higher order conceptual development in which connectionist models *have* been constructed. I turn to these next.

### Connectionist Models for the Balance Scale

McClelland (1989, 1995) noted that, although the production system models for the balance scale provided a good description of the four rules discussed earlier, they tell us little about the forces that drive children from one rule system to the next. In addition, none of the existing rule-based models can account for the torque-difference effect; thus children do better when the discrepancy between the torques on each side of the balance scale is increased (Ferretti & Butterfield, 1986; Wilkening & Anderson, 1982). McClelland constructed a back propagation model of the balance beam problem with 20 input units. One positional unit was devoted to each of the 10 pegs (5 to the left and 5 to the right of the fulcrum). Ten weight units represented 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. Every possible problem could be encoded with only 4 units turned on. For example, in a problem with 4 weights on the third peg from the right and 5 weights on the second peg from left, the units turned on would then be 4-right-weight, 5-left-weight, 3-right-distance, and 2-left-distance. To capture the common assumption that children have more exposure to weight as a cause of going-down effects, McClelland biased the network toward reliance on the weight cue over the distance cue by including a large number of cases in which the distance cue was neutralized. (This kind of hand-wired bias is justifiably used to put the model in the same initial state as the children studied in Siegler's original studies. It makes no attempt to account for how children reach this initial state.)

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 Model I and moved on to learn Model II and then Model III. It never acquired full use of Model IV, because, McClelland argued, some aspects of the use of Model IV by adults involved the application of full mathematical analysis. The network was, however, able to capture aspects of the torque distance effect mentioned previously. Torque distance effects indicate that subjects did not simply apply an all-or-none rule, but performed a cue weighting that is much like that conducted inside a neural network.

Shultz et al. (1995) extended McClelland's model by using the cascade correlation procedure described earlier. Shultz et al. argued that static back propagation 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 Models I, II, and aspects of III, but was unable to learn Model IV. Using the cascade correlation framework, however, 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 showed that connectionist models can provide good accounts of perceptual aspects of learning such as the torque distance effect. Second, as Mareschal and Shultz (1996) pointed out, cascade correlation models are inherently generative and thus provide a strong existence proof for the plausibility of a constructivist approach to cognitive development.

### COMPUTATIONAL MODELS OF OTHER DEVELOPMENTAL PHENOMENA

I have focused on the balance scale in order to compare the two approaches to computational modeling, but many other computational models now address a variety of other domains and the issue of relevance to cognitive

development. The domains include classic Piagetian tasks (conservation, seriation, object permanence) as well as arithmetic and language acquisition. The developmental issues include rule learning, stages, strategy change, generalization, and efficiency (see Table 6.3). Of particular interest is Shultz's (1997) recent cascade correlation model of number conservation, which captures an impressive array of conservation phenomena and proposes a novel explanation for some of them.

## CONCLUSION

In concluding, I want to make three points. First, the two computational approaches are not as distinct as their practitioners have often claimed (MacWhinney, 1993, makes a similar point). Second, for all of their accomplishments, both approaches must solve some very difficult remaining problems, but these problems are fairly well defined, so that progress (or failure) can be measured. Third, I suggest how to relate these new ideas to earlier Piagetian notions.

TABLE 6.3  
Recent Computational Models of Developmental Relevance

<i>Model Type Domain</i>	<i>Issues Addressed</i>	<i>Authors</i>
<i>Connectionist</i>		
Seriation	Rule learning, stages, perceptual effects	Shultz et al. (1995)
Distance, time		
Causal reasoning		
Pronoun acquisition		
Language acquisition	Learning gender of definite articles in German	MacWhinney et al. (1989)
Object permanence	Graded representations of knowledge and strategies	Munakata, McClelland, Johnson, & Siegler (1997)
Number conservation	Problem size effect, length bias effect, screening effect	Shultz (1997)
<i>Production System</i>		
Transitivity	Strategy change, capacity, complexity	Halford et al. (1995)
Number conservation	Analogical reasoning in skill acquisition	
	Durability and robustness of learning, generalization, operational level and structural change, speed of learning	T. J. Simon & Klahr (1995)
Arithmetic	Development of increasingly efficient procedures for single-digit addition	Neches (1987)
Concept learning and language acquisition	First language acquisition	Langley (1987)

### Comparing Production Systems and Connectionist Systems

What are the fundamental differences between production system and connectionist approaches? Here are some candidates:

1. *Parallelism*. The inherent parallelism of connectionist models is often contrasted with the serial recognize-act cycle of production systems. Production systems, however, also have a high degree of parallelism because during the *match* or *recognize* phase of a production system's recognize-act cycle, the condition side of all productions is matched in parallel with all the active declarative memory elements.

2. *Distributed knowledge*. The extent to which knowledge is distributed or modularized in a production system depends entirely on the grain size that elements or productions are supposed to capture. A 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. Similarly, in parallel distributed processing (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*. Nothing inherent in either formulation specifies what this grain should be until additional constraints are imposed on the model.

3. *Continuity*. Another purported difference between PDP models and production system models is the *gradualism* of the former and the *abruptness* of the latter. We can, however, create a production system architecture with continuously varying strengths of productions. Hence production systems can exhibit gradualism. Because of the appropriate grain size on a performance window, connectionist models could appear to be undergoing discontinuous changes.

Of course, there *are* important pragmatic and theoretical differences here,<sup>4</sup> but I believe that the internecine battle between the symbolic and subsymbolic camps has overstated the differences and ignored the fact that the two approaches share many important properties. Indeed, it should not be surprising that there are many points of convergence, because both approaches pursue common goals and face a common constraint: the real behavior of real children. As I noted earlier, perhaps the most important

---

<sup>4</sup>Perhaps the difference in these approaches is in the rhetoric. Although production system descriptions are burdened with a pedestrian terminology of "matching," "recognizing," "acting," "conflict resolution," and "chunking," connectionist models enjoy the lyrical characterizations of such things as "cascade correlation," "graceful degradation," "optimal harmony," "victory," and "epochs." It's hard to beat that!

common feature is the conviction that computational models provide a very precise language in which to describe the conceptual habitat.

### Problems to Be Solved

The area of unsolved problems is exciting, productive, and cumulative because the discipline of creating computational models forces our ignorance to the forefront. The unresolved questions are sufficiently specific that it is possible to assess theoretical progress (see Mareschal & Shultz, 1996, for a cogent example). A recent compendium of computational models of cognitive development (T. J. Simon & Halford, 1995) contains several illuminating disagreements among people who have modeled the same domain, but from different approaches.

It is clear that the ultimate understanding of transition mechanisms requires insights from both connectionist and production system perspectives. Claims for the superiority of one approach over the other are premature, and both approaches still face some difficult challenges. Here are a few of the issues on the research frontier for computational models of conceptual development.

*Scalability.* To date, both symbolic and subsymbolic models of cognitive development have focused on highly circumscribed domains, and in those domains, on small-scale exemplars of the domain. For all 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, Leinbach, Taraban, & McDonald, 1989). The toy model approach often reduces large problems such as question answering (St. John, 1992) or word sense disambiguation (Harris, 1994) 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 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 the work so far has been on toy versions of larger domains.

Computational modelers have argued, either explicitly or implicitly, that in principle, such models can be expanded substantially with no major theoretical modifications. But can they? The plausibility of these claims varies according to the approach, and the symbolic models have 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 artificial-intelligence-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 connectionist language models, their performance begins to degenerate. One of the major challenges for computational modelers, 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 children receive. 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 document both the language input to children and children's developing conversational competence. These data are now 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 children's productions. Increasingly, simulations of language learning are being based on properties of 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. 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.

### **Cabbages and Kings**

Finally, I want to move to a metatheoretical issue, which concerns the way in which workers in our field have viewed theoretical progress. In particular, how are the questions addressed here related to Piaget's efforts to characterize the developmental process?

The message I have attempted to convey is twofold: Questions about the conceptual habitat can be answered in terms of computational models, and an active field of research in cognitive science is exploring the capacity and limits of different cognitive architectures. The field is lively, somewhat contentious, and highly technical. I have tried to indicate its current contributions and its potential for our area, as well as some of its knottiest problems. I am concerned, however, that, as psychologists interested in cognitive development, we have been unnecessarily burdened by the shadow of the massive theoretical edifices of the past. The problem, as depicted in Fig. 6.5, is that the earlier constructs of assimilation and accommodation may impose an unnecessary and potentially unproductive constraint on both new empirical work and new theoretical concepts. Indeed, developmentalists of all stripes—including computational modelers—seem to feel obliged to comment on the extent to which their theories can be placed in correspondence with the Piagetian notions of assimilation and accommodation. For example, consider the mapping by Shultz et al. (1995):

Using Piaget's terms, one can conceptualize three general types of cognitive encounters in cascade-correlation nets: (1) assimilation, (2) assimilative learning, and (3) accommodation. Pure assimilation occurs without learning. It is represented in cascade-correlation by correct generalization to novel problems without either weight changes or hidden unit recruitment. Assimilative learning occurs by weight adjustment, but without hidden unit recruitment. Here the network learns new patterns that do not require non-linear changes in representational power. Accommodation occurs via hidden unit recruitment when new patterns cannot be learned without non-linear increases in computational power. (p. 53)

Shultz et al. go on to discuss other types of computational models in relation to the processes of assimilation and accommodation:

Adaptation through assimilation and accommodation can also be re-interpreted through rule-based and back-propagation perspectives, but with less



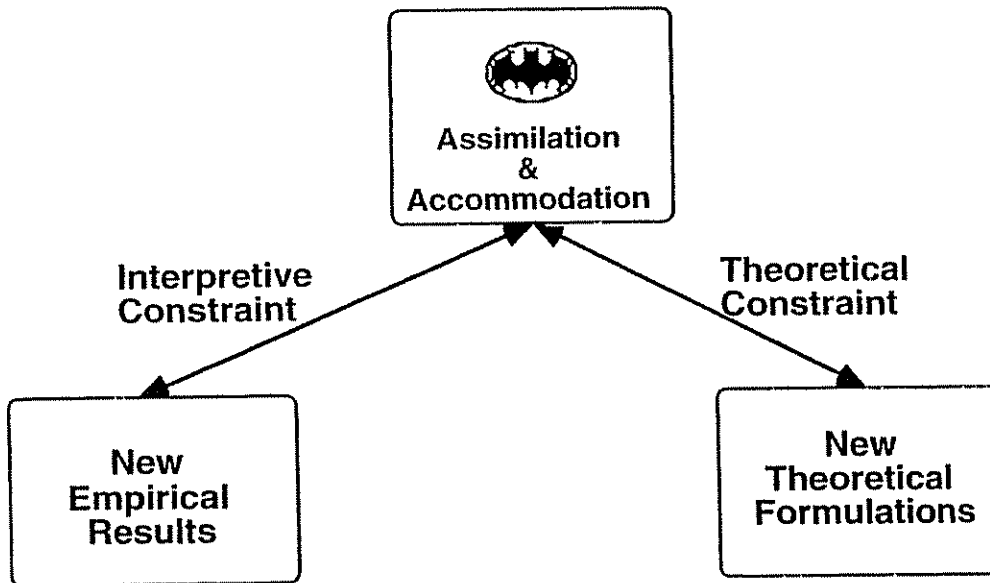


FIG. 6.5. Current neo-Piagetian paradigm.

satisfactory results. In a rule-based learning system like Soar, assimilation could be construed as rule-firing and accommodation could be construed as chunking new rules through impasse-driven search. In back-propagation learning, accommodation could be viewed in terms of weight adjustment and assimilation as the absence of such adjustment. (p. 54)

These attempts to map the new computational constructs to precomputational theoretical constructs have been widespread among connectionists. The following from Bechtel and Abrahamsen's (1991) introductory text on the topic illustrates the genre:

Connectionism could be viewed as a modern mechanism for achieving stage-like states by means of the heretofore somewhat mysterious processes of accommodation and assimilation. Specifically, assimilation can be interpreted in terms of the tendency of an interactive network to settle into the most appropriate of its stable (attractor) states . . . when input is presented to it; in Piaget's language, this is the schema to which the experience has been assimilated. Accommodation can be interpreted as the changes in activations as well as weights that occur in order to assimilate the experience. (That is, transient state changes and learning are highly interrelated both in connectionist networks and in Piaget's notion of accommodation. The assimilation of any experience involves both of these aspects of accommodation.) (p. 271)

The proclivity to look over one's theoretical shoulder for evidence of "equilibratory correctness" is not limited to computational modelers. For

example, in summarizing the current state of theory-theory, Gopnik (1996) made the mapping as follows: "Thus, the interpretive effects of theories seem much like assimilation, and the processes of falsification and counter evidence, which lead to theory change, are reminiscent of accommodation" (p. 221).

It strikes me that the search for assimilation and accommodation in modern computational theories of development represents a curiously non-Piagetian approach to conceptual development. Let me make the point by starting with a quotation from Piaget, who put it this way: "A rabbit that eats a cabbage doesn't become a cabbage; it's a cabbage that becomes rabbit—that's assimilation" (Piaget, quoted in Bringuier, 1980, p. 42). Let me make an analogy to this rabbit-cabbage relation. In this analogy, the cabbage is Piaget's theory of equilibration, and the rabbit—the entity doing the assimilation and accommodation—is us: the collective understanding of our field about the nature of cognitive development.

If our field's conceptual development followed the Piagetian model, then, as the rabbit did to the cabbage, we would assimilate and accommodate his theory. The field would, at first, accommodate its earlier theoretical constructs such that it could come to grips with new ideas. Simultaneously, the assimilation process would exercise its function, and the theoretical insights would be dissolved, decomposed, extracted, and intermingled with our existing conceptual structures. Moreover, new data, new questions, and new theoretical languages would, in their turn, be assimilated and accommodated into the conceptual structure of the field. In other words, we rabbits would digest this theoretical cabbage, would eat other cabbages and other vegetables, but would remain rabbits.

But I think something else has happened, at least in part of our field. The accommodation process is all that ever got started. To take apart the theory, to extract its essential nutrients, and pass on the rest is sometimes viewed as a misguided mixture of heresy and ignorance. But such a view, however, minimizes assimilation, and if there is no assimilation, then instead of the cabbage becoming a rabbit, the rabbit becomes a cabbage.

However, this is quite unnecessary. There is no burden of responsibility for computational modelers—or any other contemporary theorists—to scrutinize their models to identify the parts that are doing accommodation and the parts that are doing assimilation. It is hard to see how such efforts can be productive in view of the inherent ambiguity of the initial constructs. This point has been noted repeatedly in the literature:

Piaget's particular models of equilibration represent his efforts to [produce a theory of self-organization], but they fall somewhat short because of their excessive abstractness. So the task of producing a concrete theory of cognitive development as a self-organizing process remains, and that theory may or may not resemble Piaget's own models very closely. (Chapman, 1992, p. 47)

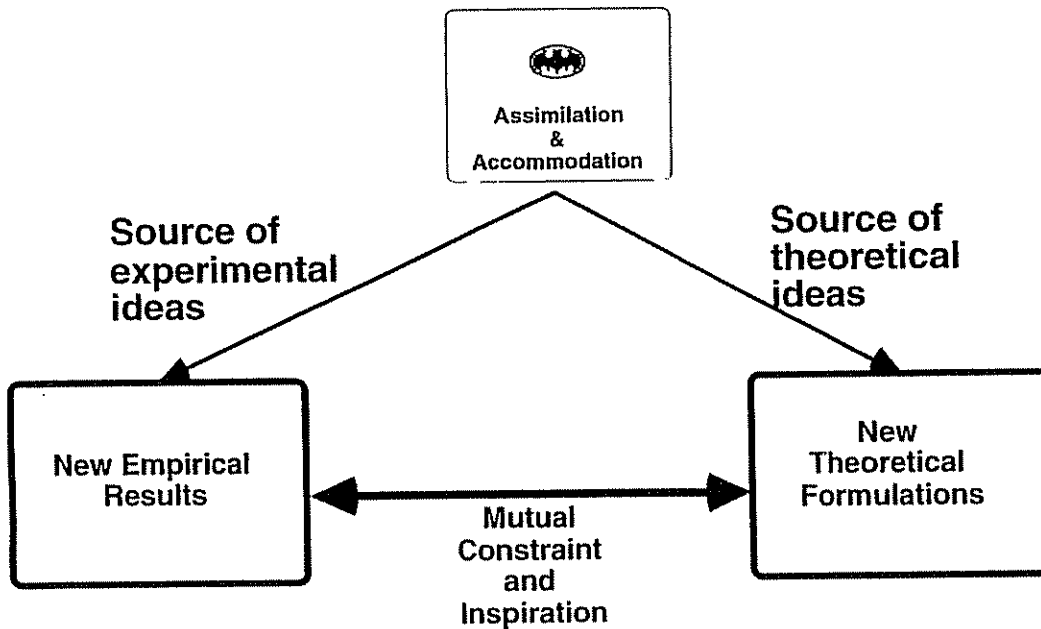


FIG. 6.6. Proposed neo-neo-Piagetian paradigm.

Theoretical development could be greatly stimulated if less effort were devoted to testing Piaget's theory and more were devoted to testing contemporary theories. This is analogous to what is done in other research areas. For example, memory researchers do not devote most of their efforts to testing theories by James (1890) or Bartlett (1932) but to contemporary theories such as those of Craik and Lockhart (1972) or Murdock (1982). Reference is still made to earlier works, but as a source of insight and ideas rather than as explicit theory. (Halford, 1989, p. 351)

This perspective suggests that it is more productive to use the old constructs as inspiration rather than as constraints (see Fig. 6.6). Our only constraints need then be between our developing theories and our developing, emerging results. Moreover, I believe that such an approach is entirely consistent with Inhelder's advice: "Instead of praising Piaget for what he accomplished, the best tribute we can pay to his memory is to go forward" (Inhelder, 1992, p. xiii).

## ACKNOWLEDGMENTS

Thanks to my colleagues Martha Alibali, Zhe Chen, and Bethany Rittle Johnson for comments on an earlier draft of this chapter. Its preparation was supported in part by a grant from the National Institute of Child Health and Human Development (R01-HD25211). Sections of this chapter

are partly based on chapters with Christian Schunn (Schunn & Klahr, 1997) and Brian MacWhinney (Klahr & MacWhinney, 1997), but I am solely responsible for the views expressed here.

## REFERENCES

- Adolph, K. (1995). Psychophysical assessment of toddlers' ability to cope with slopes. *Journal of Experimental Psychology*, *21*, 734–750.
- 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.
- Anderson, J. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Bartlett, F. C. (1932). *Remembering: A study in experimental and social psychology*. Cambridge, England: Cambridge University Press.
- 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.
- Bidell, T. R., & Fischer, K. W. (1994). Developmental transitions in children's early on-line planning. In M. M. Haith, J. B. Benson, R. J. Roberts, & B. F. Pennington (Eds.), *The development of future-oriented processes* (pp. 141–176). Chicago: University of Chicago Press.
- Bringuier, J. (1980). *Conversations With Jean Piaget*. Chicago: University of Chicago Press.
- Brown, A. L. (1982). Learning and development: The problem of compatibility, access and induction. *Human Development*, *25*, 89–115.
- Brown, R. (1970). Introduction. In Society for Research in Child Development (Ed.), *Cognitive development in children* (pp. ix–xii). Chicago: University of Chicago Press.
- Carey, S. (1985). *Conceptual change in childhood*. Cambridge, MA: Bradford Books/MIT Press.
- 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: Lawrence Erlbaum Associates.
- Case, R. (1997). The development of conceptual structures. In D. Kuhn and R. S. Siegler (Eds.), *Handbook of child psychology* (5th ed.), Vol. 2: *Cognition, perception, and language* (pp. 745–800). New York: Wiley.
- Chapman, M. (1992). Equilibration and the dialectics of organization. In H. Beilin, & P. Pufall (Eds.), *Piaget's theory, prospects and possibilities* (pp. 135–176). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Cottrell, G., & Plunkett, K. (1991). Learning the past tense in a recurrent network: Acquiring the mapping from meaning to sounds. In *Proceedings of the 13th annual conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- 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.
- Doorenbos, R. B. (1995). *Production matching for large learning systems*. Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA.
- Fausett, L. (1994). *Fundamentals of neural networks*. Englewood Cliffs, NJ: Prentice-Hall.
- Ferretti, R. P., & Butterfield, E. C. (1986). Are children's rule assessment classifications invariant across instances of problem types? *Child Development*, *57*, 1419–1428.
- Gelman, R. (1982). Accessing one-to-one correspondence: Still another paper about conservation. *British Journal of Psychology*, *73*, 209–220.
- Gopnik, A. (1996). The post-Piaget era. *Psychological Science*, *7*, 221–225.

- Halford, G. S. (1989). Reflections of 25 years of Piagetian cognitive developmental psychology, 1963–1988. *Human Development*, 32, 325–357.
- Halford, G. S. (1993). *Children's understanding: The development of mental models*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- 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* (pp. 77–156). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Harris, C. (1994). Coarse coding and the lexicon. In C. Fuchs & B. Victorri (Eds.), *Continuity in linguistic semantics* (pp. 205–229). Amsterdam, The Netherlands: John Benjamins.
- Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the theory of neural computation*. New York: Addison-Wesley.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
- 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.
- Inhelder, B. (1992). Foreword. In H. Beilin & P. Pufall (Eds.), *Piaget's theory: Prospects and possibilities* (pp. iv–x). Hillsdale, NJ: Lawrence Erlbaum Associates.
- James, W. (1890). *The principles of psychology*. New York: Holt.
- Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, 99, 122–149.
- Karmiloff-Smith, A. (1992). *Beyond modularity: A developmental perspective on cognitive science*. Cambridge, MA: MIT 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., pp. 273–336). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Klahr, D., & MacWhinney, B. (1997). Information processing. In D. Kuhn & R. S. Siegler (Eds.), W. Damon (Series Ed.), *Handbook of child psychology (5th ed)*, Vol. 2: *Cognition, perception, and language* (pp. 631–678). New York: Wiley.
- 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: Lawrence Erlbaum Associates.
- Kuhn, D. (1995). Microgenetic study of change: What has it told us? *Psychological Science*, 6, 133–139.
- Kuhn, D., Garcia-Mila, M., Zohar, A., & Andersen, C. (1995). Strategies of knowledge acquisition. *Monographs of the Society for Research in Child Development*. Vol. 60 (3, Serial no. 245).
- Langley, P. (1987). A general theory of discrimination learning. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learning and development* (pp. 99–162). Cambridge, MA: MIT 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: Lawrence Erlbaum Associates.
- Lewis, C. (1978). *Production system models of practice effects*. Ann Arbor: University of Michigan Press.
- 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: Lawrence Erlbaum Associates.
- MacWhinney, B., & Leinbach, J. (1991). Implementations are not conceptualizations: Revising the verb learning model. *Cognition*, 29, 121–157.

- MacWhinney, B. J., Leinbach, J., Taraban, R., & McDonald, J. L. (1989). Language learning: Cues or rules? *Journal of Memory and Language*, 28, 255–277.
- Marcus, G., Ullman, M., Pinker, S., Hollander, M., Rosen, T., & Xu, F. (1992). Overregularization in language acquisition. *Monographs of the Society for Research in Child Development*, 57(4).
- Mareschal, D., & Shultz, T. R. (1996). Generative connectionist networks and constructivist cognitive development. *Cognitive Development*, 11, 571–603.
- 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* (pp. 157–204). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Miller, P. H. (1983). *Theories of developmental psychology*. San Francisco: Freeman.
- Munakata, Y., McClelland, J. L., Johnson, M. J., & Siegler, R. S. (1997). Rethinking infant knowledge: Toward an adaptive process account of successes and failures in object permanence tasks. *Psychological Review*, 104, 686–713.
- Murdock, B. B., Jr. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review*, 89, 609–626.
- Neches, R. (1987). Learning through incremental refinement of procedures. In D. Klahr, P. Langley, & R. Neches (Eds.), *Production system models of learning and development* (pp. 163–220). Cambridge, MA: MIT Press.
- Newell, A. (1990). *A unified theory of cognition*. Cambridge, MA: Harvard University Press.
- Piaget, J. (1985). *The equilibration of cognitive structures* (Trans. I. Brown & J. Thampy). Chicago: University of Chicago Press. (Original work published 1975)
- Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In *Parallel distributed processing, Vol. 2: Psychological and biological models* (pp. 216–271). Cambridge, MA: MIT Press.
- Schunn, C. D., & Klahr, D. (1998). Production systems: Views on intelligent behavior. In W. Bechtel & G. Graham (Eds.), *A companion to cognitive science*. Oxford, England: Blackwell.
- Shultz, T. R. (1997). *A computational analysis of conservation*. Working paper, Department of Psychology, McGill University, Montreal, Canada.
- Shultz, T. R., 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* (pp. 205–262). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology*, 8, 481–520.
- Siegler, R. S. (Ed.). (1978). *Children's thinking: What develops?* Hillsdale, NJ: Lawrence Erlbaum Associates.
- Siegler, R. S. (1996). *Emerging minds*. New York: Oxford University Press.
- Siegler, R. S., & Crowley, K. (1991). The microgenetic method: A direct means for studying cognitive development. *American Psychologist*, 46, 606–620.
- 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: Lawrence Erlbaum Associates.
- Simon, D. P., & Simon, H. A. (1978). Individual differences in solving physics problems. In R. Siegler (Ed.), *Children's thinking: What develops?* (pp. 325–348). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Simon, H. A. (1962). An information processing theory of intellectual development. *Monographs of the Society for Research in Child Development*, 27 (Serial No. 82).
- Simon, T. J., & Halford, G. (Eds.) (1995). *Developing cognitive competence: New approaches to process modeling*. Hillsdale, NJ: Lawrence Erlbaum Associates.

- Simon, T. J., & Klahr, D. (1995). A theory of children's learning about number conservation. In T. J. Simon & G. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 315–354). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Spensley, M. F. (1995). *Representational redescription and the development of cognitive flexibility*. Unpublished doctoral dissertation, The Open University, London, England.
- St. John, M. (1992). The story gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science*, *16*, 271–306.
- Thelen, E., & Smith, L. (1994). *A dynamic systems approach to the development of cognition and action*. Cambridge, MA: MIT Press.
- van der Maas, H., & Molenaar, P. (1992). Stagewise cognitive development: An application of catastrophe theory. *Psychological Review*, *99*, 395–417.
- 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* (pp. 359–435). Cambridge, MA: MIT Press.
- 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: Lawrence Erlbaum Associates.