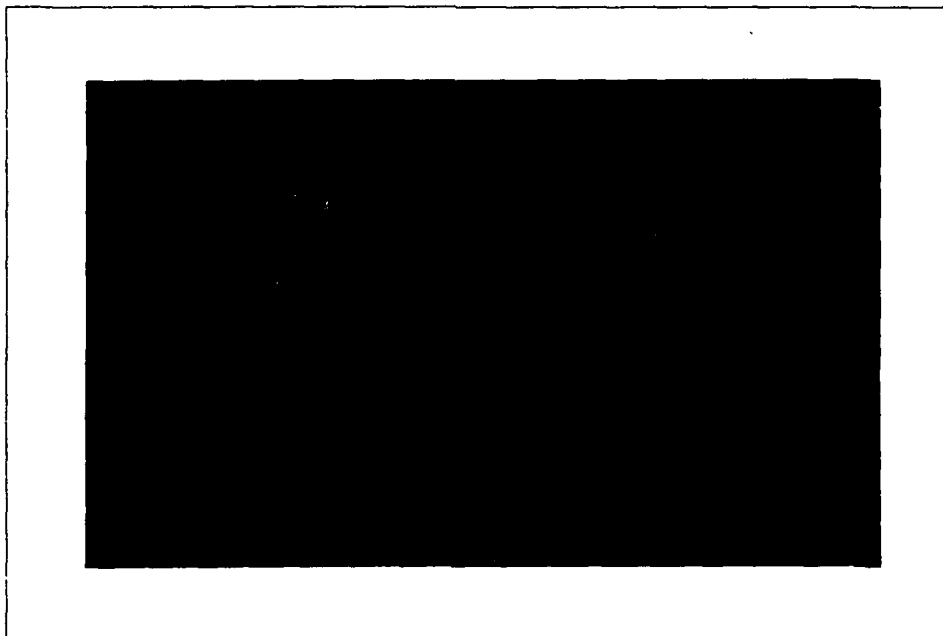


DTIC FILE COPY

AD-A219 039



The Artificial Intelligence and Psychology Project

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

DTIC
ELECTE
MAR 14 1990
S B D

Approved for public release; distribution unlimited.

90 03 12 069

(2)

. . .

**INFORMATION PROCESSING APPROACHES
TO
COGNITIVE DEVELOPMENT**

Technical Report AIP - 42

David Klahr

Carnegie Mellon University
Department of Psychology
Pittsburgh, Pa., 15213

1 July 1988

DTIC
S **ELECTE** **D**
MAR 14 1990
B

I would like to thank my colleagues Janet Davidson, Sharon Carver and Robert Siegler for their careful and constructive comments on an earlier version of this paper. This research was supported by the Computer Sciences Division, Office of Naval Research and DARPA under Contract Number N00014-86-K-0678 and Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research under Contract No. N00014-86K-0349. Reproduction in whole or in part is permitted for purposes of the United States Government. Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP - 42			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Carnegie-Mellon University		6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION Computer Sciences Division Office of Naval Research	
6c. ADDRESS (City, State, and ZIP Code) Department of Psychology Pittsburgh, Pennsylvania 15213			7b. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, Virginia 22217-5000		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as Monitoring Organization		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0678	
8c. ADDRESS (City, State, and ZIP Code)			10. SOURCE OF FUNDING NUMBERS p4000ub201/7-4-86		
PROGRAM ELEMENT NO N/A		PROJECT NO N/A		TASK NO N/A	
				WORK UNIT ACCESSION NO N/A	
11. TITLE (Include Security Classification) Information processing approaches to cognitive development					
12. PERSONAL AUTHOR(S) Klahr, David					
13a. TYPE OF REPORT Technical		13b. TIME COVERED From 86Sept15 to 91Sept14		14. DATE OF REPORT (Year, Month, Day) 1988 July 1	
15. PAGE COUNT 52					
16. SUPPLEMENTARY NOTATION To appear in R. Vasta (Ed.), Annals of child development, vol. 6, 1989.					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	information-processing, children's thinking		
			cognitive development		
			computer simulating. (EE)		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) SEE REVERSE SIDE					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION		
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Alan L. Meyrowitz			22b. TELEPHONE (Include Area Code) (202) 696-4302		22c. OFFICE SYMBOL N00014

Cont From
Reverse Side
01/14/93

ABSTRACT

This chapter reviews the history and current status of information-processing approaches to cognitive development. Because the approach is so pervasive, it is useful to characterize research in terms of distinctive features, and to organize the features according to whether they are 'soft-core' or 'hard-core' aspects of the information processing approach, as follows:

Features of soft-core information processing approaches:

THEORETICAL FEATURES

- S1 The assumption that the child's mental activity can be described in terms of processes that manipulate symbols and symbol structures
- S2 The assumption that these symbolic processes operate within an information processing system with identifiable properties, constraints, and consequences
- S3 The characterization of cognitive development as self-modification of the information processing system

METHODOLOGICAL FEATURES

- S4 Use of formal notational schemes for expressing complex, dynamic systems
- S5 Modelling the time-course of cognitive processing over relatively short durations: chronometric analysis
- S6 Use of high-density data from error-patterns and protocols to induce and test complex models.
- S7 Use of highly detailed analyses of the environment facing the child on specific tasks

Features of hard-core information processing approaches:

- H1 Use of computer simulation
- H2 Commitment to elements of the simulation as theoretical assertions rather than just metaphor or computational convenience
- H3 Goal of creating a complete self-modifying simulation that accounts for both task performance and development.

Each of these features is illustrated by example, and the hard-core approach is expanded into a detailed analysis of self-modifying production systems and their potential for formulating theories of cognitive development.

9147540
F

Table of Contents

1 Characterizing information-processing approaches	1
1.1 Soft-core information-processing approaches to cognitive development	3
1.1.1 Theoretical assumptions	3
1.1.2 Methodological practice	5
1.1.3 Topic areas and subject populations	12
1.1.4 Soft-core information processing: What's not included?	13
1.2 Hard-core information-processing approaches to cognitive development	13
1.3 H1: Use of computer simulation	14
1.3.1 The computer's role in simulation models	17
1.3.2 Recursive decomposition and emergent properties	18
1.3.3 Data constraints	19
1.4 H2: Commitment to elements of the simulation as theoretical assertions, rather than just metaphor or computational convenience	20
1.5 H3: Goal of creating a complete self-modifying simulation that accounts for both task performance and development	21
2 Production systems: At the core of the core	
2.1 Notation or theory?	22
2.2 Properties of production-system models	23
2.3 Production systems as cognitive architectures	23
2.3.1 Working memory issues	24
2.3.2 Production memory issues	24
2.3.3 Conflict resolution issues	25
2.4 Some examples of production-system models of children's performance	25
2.4.1 Quantification: Matching production firings to chronometric data	25
2.4.2 Production systems for different levels of performance	26
2.4.3 Representing the immediate task context	27
2.4.4 Multiple-level production system: From principles to encodings	28
2.4.5 Non-transition models: A summary	28
2.5 Self-modification	29
2.5.1 Mechanisms	29
2.5.2 Change during the match	30
2.5.3 Change during conflict resolution	30
2.5.4 Changing conditions and actions	31
2.5.5 Are other mechanisms necessary?	32
2.5.6 Heuristic procedure modification	33
2.6 Summary: Production systems as frameworks for cognitive developmental theory	33
3 Conclusion and speculation	34
3.1 Is this trip necessary?	35
3.2 Connectionism and cognitive development	36
3.3 Self-modifying systems: Development or learning?	36
3.4 The future of the hard-core approach	38

n For

A&I

ed

tion

By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



1 Characterizing information-processing approaches

Reflections on the intellectual history of a field often reveal a long period between the occurrence of fundamental insights and the first concrete steps based on those insights. Over 25 years ago, Herbert Simon (1962) suggested the general form of an information-processing approach to cognitive development:

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].

This provocative idea motivated my own early research with Iain Wallace (cf. Klahr & Wallace, 1970a, 1970b, 1972), but not until ten years after Simon's suggestion did an entire volume explicitly focused on "Information Processing in Children" (Farnham-Diggory, 1972) appear. The chapters in that book represent an interesting contrast between traditional approaches to perception and memory (e.g., Pollack, 1972; Hagen, 1972), Geneva views on information-processing issues (Inhelder, 1972; Cellier, 1972), and important considerations surrounding information-processing approaches to development (Newell, 1972).

A few years later, when Iain Wallace and I were writing a monograph entitled "Cognitive Development: An Information Processing View" (Klahr and Wallace, 1976), we chose the indefinite article in our title carefully. The field of adult information-processing psychology was expanding rapidly and diffusely, and we were well aware that our view of important issues and proposed solutions was neither comprehensive nor representative. Indeed, we believed that, with respect to adult cognition, there was no single perspective that could characterize the entire field of information processing, and therefore no single vantage point from which to present the information-processing view of cognitive development.

With the passage of another dozen years, the definitional task has become no easier. The very pervasiveness of information-processing psychology contributes to the difficulty, and the imperialism implicit in some definitions exacerbates it. Another problem in deciding what is and is not an example of the information-processing approach is that, "many developmental psychologists ... are not aware that they have accepted certain assumptions and methods of the information-processing approach" (Miller 1983, p. 249). Further complicating the problem is the fact that many others have already trod this ground and have offered their own definitions of information-processing psychology in general (e.g., Lachman, Lachman & Butterfield, 1979; Palmer & Kimchi, 1986) and information-processing within the field of cognitive development (e.g., Bisanz et al., 1987; Siegler, 1983; Neches, 1982; Rabinowitz et al., 1987; Klahr & Wallace, 1976). Nevertheless, in this chapter I will accept the challenge presented by the Editor of this volume and attempt to say something about "information processing" that may be useful to readers of *Annals of Developmental Psychology*. In so doing, I will offer some personal

opinions about the nature of the field, and I will sample from and respond to previous definitions and criticisms.

Few people would disagree with the recent claim that, with respect to alternative approaches for understanding *adult* cognition:

the one that has dominated psychological investigation for the last decade or two is information processing. For better or worse, the information-processing approach has had an enormous impact on modern cognitive research, leaving its distinctive imprint on both the kinds of theories that have been proposed and the kinds of experiments that have been performed to test them. Its influence has been so pervasive, in fact, that some writers have argued that information processing has achieved the exalted status of a "Kuhnian paradigm" for cognitive psychology (Lachman, Lachman & Butterfield, 1979). It is unclear whether or not this claim is really justified, but the fact that it has even been suggested documents the preeminence of information processing in modern cognitive psychology. (Palmer & Kimchi, 1987, p. 37)

Deciding whether information processing is equally preeminent in cognitive development depends in large part on how far one chooses to cast one's definitional net. The broadest definitions of information-processing approaches to cognitive development usually invoke the family resemblance concept: An approach qualifies for inclusion to the extent that it manifests a certain set of features. Although no single approach uses all of them, the more features that are present in a piece of work, and the more highly articulated those features, the more typical it is of the approach.

It will be convenient in this paper to propose a dichotomy between "hard core" and "soft core" information-processing approaches, based on the set of features that they exhibit. To preview the set of features that will be used, I have listed them all in Table 1; they will be elaborated in subsequent sections. The hard/soft distinction serves to organize this paper, but the terms should not be viewed as mutually exclusive. In fact, all of the soft core features can be mapped into their stronger versions in the hard core set. The mapping will become evident as the features are described. It will also become evident that the universality of information-processing to which Palmer and Kimchi refer applies only to the soft-core approaches, while the hard core, as it will be defined here, applies to a relatively small, but influential, part of the field.

The chapter is organized as follows. In the remainder of this section, I characterize the defining features of information-processing approaches to cognitive development. This will include a sample of illustrative instances. Section 2 describes a particular information-processing approach -- one based on production-system models -- that is becoming very influential. Finally, Section 3 summarizes what I think the major accomplishments have been so far, and some speculation about future directions

Insert Table 1 about here

1.1 Soft-core information-processing approaches to cognitive development

The features that characterize soft-core approaches can be grouped into two categories: theoretical assumptions and methodological practices¹.

1.1.1 Theoretical assumptions

S1: The child's mental activity can be described in terms of processes that manipulate symbols and symbol structures. My use of the terms "symbol" and "symbol-structure" here is quite distinct from the symbolic thought associated with ideas such as Vygotsky's "symbolic play" or Piagetian questions about when a child makes a transition from pre-symbolic to symbolic functioning. Symbolization in that diffuse sense concerns the general issue of the power of the child's representational capacity, not whether or not symbols are involved. Instead, I am using symbols at a more microscopic level, in the sense intended by Newell (1980), where symbols provide access to other symbols. Such symbols comprise the elementary units in any representation of knowledge including sensory-motor knowledge or linguistic structures. Thus, distinctions between dense and articulated symbols (Goodman, 1968) or personal and consensual symbols (Kolars and Smythe, 1984) are not relevant at the level of underlying symbols necessary to support all symbolic capacity. Given this microscopic interpretation of what a symbol is, it seems to me that the symbolic assumption is so deeply embedded in the field that often it is only implicit, and its use ranges from interpretations of relatively focused studies to all-encompassing theoretical positions.

For example, DeLoache (1987) discovered an abrupt improvement between 30 and 36 months in children's ability to understand the symbolic relationship between a model of a room and the real room. She summarizes this as a milestone in "the realization that an object can be understood both as a thing itself and as a symbol of something else." (DeLoache, 1987, p.1556), and she notes that the younger children fail "to think about a symbolic object both as an object and as a symbol" (p. 1557). Thus, at the global (or conventional) level, DeLoache's results suggest that the 2.5 year-old children are "pre-symbolic" (at least on this task.) But it is clear that if one were to formulate detailed models of children's knowledge about this task at both levels of performance, then one would, in both cases, postulate systems that had the ability to process symbols at the microscopic level defined above. Thus, even in an ingenious research program -- such as DeLoache's -- directed at determining when children "become symbolic", the assumption of underlying symbol-processing capacity remains.

The second example of implicit assumptions about symbol processing comes from Case's (1985, 1986) theory. He postulates *figurative schemes*, *state representations*, *problem representations*, *goals*, *executive control structures*, and *strategies* in order to account for performance at specific levels of development, and *search*, *evaluation*, *retagging*, and *consolidation* to account for development from one performance level to the next. Case makes no explicit reference to *symbol structures*, but his central theoretical construct -- what he calls *Short Term Storage Space* (STSS) -- clearly implies that the kinds of things that get processed are comprised of symbols and symbol

¹For other recent definitions of the field, see Kail & Bisanz, 1982 and Siegler, 1983, 1986. For a thoughtful comparison between information processing and other major approaches to cognitive development, such as Piagetian, Freudean, Gibsonian, see Miller, 1983.

structures. Thus, although Case commonly contrasts his own approach (cf. Case, 1985, pp. 43-50) with hard-core information-processing approaches that rely on computer simulation, I view his work as being well within the domain of soft-core information processing.

Explicit assumptions about the centrality of symbol structures are exemplified by the "knowledge is power" approach to cognitive development. The general goal of this line of work is to demonstrate that much of the advantage that adults have over children derives from their more extensive knowledge base in specific domains, rather than from more powerful general processes. The most convincing evidence supporting this position comes from Chi's studies (Chi, 1976, 1977, 1978) in which children who have more domain-specific knowledge than adults (e.g., children who have more knowledge about chess or dinosaurs or classmates' faces) outperform their adult counterparts on a range of tasks in which access to the specific knowledge is a determining factor in performance. In all of these, and related, studies, the major explanatory variable is access to symbolic structures (chunks, semantic nets, etc.) that supports the superior performance of the children.

S2: These symbolic processes operate within an information processing system with identifiable properties, constraints, and consequences. Typically, developmentalists interested in a variety of cognitive processes have assumed an architecture having the canonical form of the STM/LTM model of the late '60s and early '70s (cf. Atkinson & Shiffrin, 1968; Craik & Lockhart, 1972; Norman, Rumelhart, & LNR, 1975). This architecture is comprised of several sensory buffers (e.g., "iconic" memory, an "acoustic buffer", a limited capacity short-term memory, and an unlimited, content-addressable long-term memory. Newell (1972, 1973, 1981) developed the concept of *cognitive architecture* of the mind, and both he and Anderson (1983) have made very specific proposals about how it is structured. Cognitive architectures can be cast at several levels, just as one can discuss the architecture of a computer chip, or the entire central processing unit, or the micro-code, and so on, up to the architecture of a high-level user application. The cognitive architectures proposed by Newell and by Anderson pertain to the higher levels, corresponding roughly to the program interpreter level, whereas other widely accepted proposals for cognitive architectures focus on lower level issues, such as the rates and capacities of short-term memory, and the relation between short- and long-term memory.

Developmental researchers interested in higher-level problem-solving processes such as seriation, arithmetic and problem-solving (e.g., Baylor & Gascon, 1974; Neches, 1987; Klahr & Wallace, 1972; Young, 1976) have adopted a very specific form of the higher level cognitive architecture: the production system architecture proposed by Newell and Anderson. But the topic of specific architectures, such as production systems, takes us from soft-core to hard-core information processing, so I will defer that discussion until later.

Note that proposals for cognitive architectures are not the same as theories that attempt to characterize the "structure of thought". Such approaches, best exemplified by Piaget, have been recently refined and extended by such theorists as Halford (1975) and Fischer. For example, Fischer's skill theory (Fischer, 1980; Fischer and Pipp, 1984) is cast entirely in terms of abstract structures with scant attention to processes. The transition processes that he does discuss -- substitution, focusing, compounding, differentiation and intercoordination -- are presented only in terms of their global

characteristics, and are not constrained by an underlying architecture that processes information.

S3: Cognitive development occurs via self-modification of the information-processing system. This assumption shows up in several guises, ranging from Piaget's original assertions about assimilation, accommodation, and the active construction of the environment, to proposals for various kinds of structural reorganizations (e.g., Case, 1986; Halford, 1970; Fischer, 1980), to interaction between performance and learning (Siegler, 1988), to explicit mechanisms for self-modifying computer models (Klahr, Langley and Neches, 1987). This emphasis on self-modification does not deny the importance of external influences such as direct instruction, modelling, and the social context of learning and development. However, it underscores the fact that whatever the form of external environment, the information-processing system itself must ultimately encode, store, index and process that environment. Here too, the soft-core approaches tend to leave this somewhat vague and implicit, whereas the hard-core approaches make specific proposals about each of these processes. However, all information-processing approaches to development acknowledge the fundamental importance of the capacity for self-modification.

1.1.2 Methodological practice

S4: Use of formal notational schemes for expressing complex, dynamic systems. While using computer simulation languages may be *sine qua non* of hard-core information processing, there are several lesser degrees of formalization that mark the soft-core methods including such devices as scripts, frames, flow-charts, tree diagrams, and pseudo-programming languages. Compared to verbal statements of theoretical concepts and mechanisms, each of these notations offers increased precision and decreased ambiguity. Flow charts are perhaps the most common type of formal notation used by information-processing psychologists. For example, Sternberg and Rifkin (1979) used a single flow chart to represent four distinct models of analogical reasoning. Their depiction clearly indicates how the models are related and what parameters are associated with each component of each model.

Another type of formal notation commonly used in research on children's comprehension of stories is the *story grammar* (Mandler and Johnson, 1977; Stein and Glenn, 1979), and Nelson has analyzed children's event representations in terms of *scripts* (Nelson and Gruendel, 1981). Mandler (1983) provides a comprehensive summary of how these kinds of representations have been used in developmental theory. In both areas the underlying theoretical construct has been the *schema*. As Mackworth (1987) wryly notes, to simply assert that some aspect of the mind can be characterized as a schema is to say almost nothing at all, because the schema concept

has repeatedly demonstrated an ingenious talent for metamorphosis. A schema has been variously identified with a map, a record, a pattern, a format, a plan, a conservation law (and a conversation law), a program, a data structure, a co-routine, a frame, a script, a unit, and an agent. Each of these concepts has, in turn, considerable variability and ambiguity.

However, if one goes further, and makes specific proposals for how the schema is structured, organized, and processed, then this kind of formalization can be useful. For example, Hill and Arbib (1984) have attempted to clarify some of the different senses in

which 'schema' has been used, and they go on to describe a schema-based computational model of language acquisition.

The issue of how to evaluate different forms of knowledge representation is discussed at length by Klahr and Siegler (1978). They list the following criteria that a theorist could use in choosing a representation:

1. The representation must be sufficient to account for behavior. Thus, it must have a clear mapping onto the empirical base it is supposed to account for.
2. It should be amenable to multiple-level analyses. That is, it should be easy to aggregate and disaggregate the grain of explanation. For the design of well-controlled experiments or curriculum design, the representation will have to be stated in terms of averages across many subjects; it must be a model form. For detailed study of individual strategies and component processes, it must be capable of disaggregation without drastic revision.
3. The representation should not violate well-established processing constraints.
4. The representation should have 'developmental tractability' (Klahr and Wallace, 1970b). That is, it should allow us to state both early and later forms of competence and provide an easy interpretation of each model as both a precursor and successor of other models in a developmental sequence. [Klahr and Siegler, 1978, p. 65]

The attractive property of any type of formal notation is that it renders explicit what may have only been implicit, and it frequently eliminates buried inconsistencies. Siegler (1983) illustrates this point in his account of the evolution of his ideas about children's number concepts:

... I have recently adopted a more detailed representational language to characterize preschoolers' knowledge of numbers. This format involves task-specific flow diagrams operating on a semantic network; the semantic network includes the types of information that the rule models did not explicitly represent. I have had to revise my models of counting, magnitude comparison, and addition several times after I thought they were complete, because when I reformalized the ideas, the models revealed gaps and contradictions. The concreteness of the flow diagrams and semantic networks thus has added to the conceptual rigor of the ideas, forcing me to face vagueness and incompleteness in my thinking that I otherwise might have overlooked. (pp. 163-164)

What about mathematical modelling of developmental phenomena? Should it be included in the set of formal notational schemes that signal soft-core information processing? The situation is not straightforward. On the one hand, mathematical modelling meets the criteria of formalization and precision. But on the other hand, most of the mathematical models in developmental psychology typically characterize information at a very abstract level: in terms of states and transition probabilities, rather than in terms of structural organization and processes that operate on that structure (cf. Brainerd's (1987) Markov models of memory processes). As Gregg and Simon (1967)

demonstrated very clearly with respect to stochastic models of concept learning, most of the interesting psychological assumptions in such models are buried in the text surrounding the mathematics, and "the accurate predictions of fine-grain statistics that have been achieved with [stochastic theories] must be interpreted as validations of the laws of probability rather than of the psychological assumptions of the theories" (p.275). For example, Wilkinson and Haines (1987) use Markov learning models to propose some novel answers to the important question of how children assemble simple component skills into reliable strategies. However, they couch their analysis in terms the probabilities of moving between abstract states, while their discussion in the text is rife with undefined processes whereby the child "discovers", "adopts", "retains", "invokes", "moves", "prefers", "abandons", or "reverts". As is often the case in the use of mathematical models, the formalism of the mathematics obscures the informality of the underlying theory. Perhaps this is the reason why mathematical modelling has not played a central role in information-processing approaches to development.

S5: Modelling the time-course of cognitive processing over relatively short durations: chronometric analysis. Among adult experimentalists, one of the methodological hallmarks of an information-processing approach is the use of chronometric analysis. It is based on several assumptions. First, there is a set of distinct, separable, processes that underlie the behavior under investigation. Second, the particular process of interest can be isolated, via a task analysis, such that experimental manipulations can induce the system to systematically increase or decrease the number of executions of the focal process. Third, that the experimental manipulations affect *only* the number of executions of the focal process, and nothing else about that process or the total set of processes in which it is embedded. (For a thorough discussion of the history and methodology of chronometric studies, primarily with adults, see Chase, 1978.)

One of the first studies to use chronometric analysis with children was Groen and Parkman's (1972) analysis of how first graders did simple addition problems. Groen and Parkman proposed several plausible alternative models and, from each, predicted a pattern of reaction times as a function of different relations among the two addends (sum, difference, min, max). One of these models was called the "min strategy", in which subjects compute the sum by starting with the larger of the two addends and counting up the number of times indicated by the smaller of the two, producing a final result that is the sum of the two. By assuming that the initial determination of the maximum takes a fixed amount of time, this model predicts that reaction times should be a linear function of the smaller of the two arguments. Based on their analysis of mean reaction times across subjects and trials, Groen and Parkman concluded that the "min strategy" was the best fitting model. (There were some exceptions to this general result, and this process has been further elaborated with respect to individual variations across problems and subjects by Siegler (1989), and older children by Ashcraft (1982) but the initial Groen and Parkman work still stands as a pioneering effort in chronometric analysis of children's performance)

Another use of chronometric methods with children is exemplified by Keating and Bobbit's (1978) extension of Sternberg's (1966) memory-scanning paradigm. The basic task is to present children with a set of digits, followed by a "probe" digit. The child's task is to decide whether the probe digit was in the original set. Reaction time is measured from the onset of the probe until the child's responds. In addition to the general assumptions listed above, the paradigm assumes that the items in the set are stored in some kind of passive buffer, and that there is an active process that

sequentially compares the probe with each of the items stored in the buffer. The empirical question is how long each comparison (and move to the next item) takes for children at different levels of development.

Additional examples of chronometric analysis include Chi and Klahr's (1975) work on rates of subitizing and counting in 5-year olds, and Kail, Pellegrino, and Carter's (1980) study of mental rotation speeds in 9-year olds. All of these share another common feature of information-processing experiments: their goal is to go beyond testing hypotheses about some component of the cognitive system by *measuring* some of its properties. That is, the purpose of a study such as Keating and Bobbitt's is not just to demonstrate that children's memory scanning process was organized in the same way as adults', but to estimate some of the critical parameters of processes such as the scanning rate.² Kail (1988) presents an elegant example of the extent to which chronometric analysis can illuminate important developmental questions. For each of the 15 ages from 8 yrs to 22 yrs (e.g., 8-yr olds, 9-yr olds, etc.), he estimated the processing rate for five tasks: mental rotation, name retrieval, visual search, memory search and mental addition. Then he plotted the processing rate vs age function for each task, and showed that the exponential decay functions for all tasks could be fit by a single decay parameter. He interprets these results by positing an increasing amount of common, non-specific processing resources that become available to children as they develop.

S6: Use of high-density data from error-patterns and protocols to induce and test complex models. It has been often noted that pass/fail data provide only the grossest form of information about underlying processes. Nevertheless, a casual glance through the journals overflowing my in-basket reveals that most of the empirical research in cognitive development is still reported in terms of percentage of correct answers. Another characteristic of information-processing approaches is the belief that much more can be extracted from an appropriate record of children's performance. The basic assumption is that, given the goal of understanding the processing underlying children's performance, we should use all the means at our disposal to get a glimpse of those processes as they are occurring, and not just when they produce their final output. Verbal protocols, eye-movements, and error patterns (as well as chronometric methods, mentioned above) all provide this kind of high-density data.

This position is neither novel nor radical. Once again, Piaget turns up as a charter member of the soft-core information-processing club. He was probably the first to demonstrate that children's errors could reveal as much, or more, about their thought processes as their successes, and a substantial proportion of his writing is devoted to informal inferences about the underlying knowledge structures that generate children's misconceptions in many domains. Siegler (1981) puts the issue this way:

Many of Piaget's most important insights were derived from examining children's erroneous statements; these frequently revealed the type of changes in reasoning that occur with age. Yet in our efforts to make knowledge-

²In fact, Naus and Ornstein (1983) used the memory scanning paradigm to show that third-graders used a less efficient strategy than sixth-graders and adults when searching lists that could be taxonomically organized.

assessment techniques more reliable and more applicable to very young children, we have moved away from this emphasis on erroneous reasoning and also away from detailed analyses of individual children's reasoning. ... The result may have been a loss of valuable information about the acquisition process. ... [My] hypothesis is that we might be able to increase considerably our understanding of cognitive growth by devoting more attention to individual children's early error-prone reasoning. (p.3)

The basic assumption in error-analytic methodologies is that children's knowledge can be represented as a set of stable procedures that, when probed with an appropriate set of problems, will generate a characteristic profile of responses (including specific types of errors). Application of this idea to children's performance reached perhaps its most elegant form in the BUGGY models of children's subtraction errors (Brown and Burton, 1978; Brown and VanLehn, 1982). Brown and his colleagues demonstrated that a wide variety of subtraction errors could be accounted for by a set of "bugs" that children had in their subtraction procedure. For example, two of the most frequent bugs discovered by Brown & Burton were:

BORROW FROM ZERO:

When borrowing from a column whose top digit is 0,	103
the student writes 9, but does not continue borrowing	-45
from the column to the left of the zero.	158

SMALLER FROM LARGER:

The student subtracts the smaller digit in a column from	254
the larger regardless of which one is on top.	-118
	144

These and dozens of more subtle and complex bugs were inferred from the analysis of thousands of subtraction test items from 1300 children. The key to the analysis was the creation of a network of subprocedures that comprise the total knowledge required to solve subtraction problems. This procedural network can then be examined for possible points of failure, any one of which would result in a bug.

Another highly productive research program based on the analysis of error patterns is Siegler's well-known "rule assessment" methodology (Siegler, 1976; Siegler, 1981). The basic idea in this and other developmentally-oriented error-analysis work (e.g., Baylor & Gascon, 1974; Fay & Mayer, 1987; Klahr & Robinson, 1981; Young, 1976) is that children's responses at any point in the development of their knowledge about an area are based on what they know at that point, rather than on what they don't know. In order to characterize that (imperfect) knowledge, the theorist attempts to formulate a model of partial knowledge that can generate the full set of responses -- both correct and incorrect -- in the same pattern as did the child. The model thus becomes a theory of the child's knowledge about the domain at that point in her development.

Fay and Mayer (1987) extended the Brown and Burton (1978) approach from the domain of "simple" arithmetic, to the more complex domain of spatial reference in a graphics programming environment. They investigated children's naive conceptions about spatial reference by examining how children (from 9 to 13 years old) interpreted Logo

commands to move and turn from various initial orientations. Children were presented with problems that varied in initial orientation of the "turtle", the type of command (move or turn), and the value of the argument (how far to move or turn). Their task was to predict the final orientation of the turtle, given its initial orientation and command. Fay and Mayer first constructed an ideal model, comprised of about a dozen elementary operations. Then, based on the general characteristics of children's errors, they proposed six types of misconceptions (e.g., that a right-turn command actually *slides* the turtle to the right) and formulated models for the micro-structure of each misconception, in terms of degenerate versions of relevant parts of the ideal model. For the subjects to which these degenerate models were applied, Fay and Mayer were able to account for nearly every one of the (mostly) incorrect responses to the 24 items in their test battery.

Error-analyses of this type are not only useful for cognitive developmental theory, but they also have pedagogical implications. The potential for facilitating remedial instruction is what originally motivated the Buggy work, and it continues to be a valuable by-product of detailed error-analysis research:

... novice Logo programmers appear to enter the Logo environment with individual confusions and misconceptions that they apply fairly consistently during instruction. Diagnosis of the specific confusions -- such as a misunderstanding of what left and right mean or a misunderstanding of what degrees of rotation means -- provides a more detailed and potentially useful evaluation of student's knowledge than the traditional global measurement of percentage correct. (Fay & Mayer, 1987, pp?)

I believe that this kind of work illustrates the basic premise of this aspect of information-processing approaches: Careful and creative analysis of complex error patterns can provide an extremely informative window into the child's mental processes.

Protocol analysis is another form of high-density data that is often associated with information-processing approaches. The basic idea here is that in addition to final responses on tasks, the subject can generate external indications of intermediate states, and that this pattern of intermediate indicators (the protocol) can be highly informative about the underlying processes that generated the final response. Included here are not only verbal protocols, but also sequences of eye movements (Just and Carpenter, 1978) and other motor responses (Rumelhart and Norman, 1981). The classic verbal protocol analyses with adults are reported in Newell and Simon (1972), and a rigorous theoretical and methodological treatment is offered in Ericsson and Simon (1984). Here too, there is a common misconception that protocol analysis requires subjects to give an introspective account of their own behavior, and therefore is unreliable and unacceptably subjective (Nisbett and Wilson, 1977). Clearly, this would be a fatal flaw in the methodology, especially if it is to be used with children. But the criticism is unfounded. As Anderson (1987) summarizes the issue:

Many of these unjustified criticisms of protocols stem from the belief that they are taken as sources of psychological theory rather than as sources of data about states of the mind. For the latter, one need not require that the subject accurately interpret his mental states, but only that the theorist be able to specify some mapping between his reports and states of the theory. (p. 472)

In adult information-processing psychology, protocol analysis is a widespread method, but it is only infrequently used in more than a casual fashion by current cognitive developmentalists. This is very surprising, when one considers the fact that Piaget was the most prolific collector and analyser of verbal protocols in the history of psychology.

Klahr and Robinson (1981) used a combination of motor and verbal protocol analysis and error analysis to explore pre-school children's problem-solving and planning skills. Children were presented with puzzles requiring from 2 to 7 moves to solution, and they were instructed to describe the full sequence of moves that would enable them to reach the goal configuration. Children were video-taped as they described -- verbally and by pointing -- what sequence of moves they would use to solve the problem, but the pieces were never actually moved. The protocols enabled Klahr and Robinson to infer the children's internal representation of the location of each object, and the processes whereby children made moves. They then constructed several alternative models of children's strategies, and used the error-analysis technique described earlier to identify each child's response pattern with a specific strategy. Note that nowhere were the children asked to reflect on their own mental processes, or to give a report on what strategies they were using while solving the problems.

The information extracted from the protocols in the Klahr and Robinson study consisted of a planned sequence of well-defined moves of discrete objects, and this level of mapping from the protocol to hypothesized representations and processes is characteristic of the kind of protocol analyses presented in Newell and Simon's (1972) seminal work. A "richer" use of protocols, similar to some of the later examples in Ericsson and Simon (1984), provides the basis of Dunbar and Klahr's (1988) analysis of children's strategies for scientific reasoning. Children (aged 8 to 11 years old) and adults were presented with a programmable robot, taught about most of its operating characteristics, and then asked to discover how some additional feature worked. They were asked to talk aloud as they generated hypotheses, ran experiments (i.e., wrote programs for the robot and ran them), and made predictions, observations and evaluations. These verbal protocols were then analyzed in terms of different classes of hypotheses, the conditions under which experiments were run, how observed results were assessed, and so on. Based on this analysis, Dunbar and Klahr were able to suggest some important differences in scientific reasoning skills between children and adults.

S7: Use of highly detailed analyses of the environment facing the child on specific tasks. Both chronometric techniques and error analysis require at least a rudimentary analysis of the task environment. In addition, there are some information-processing approaches in which complex and detailed task analysis plays a central role, even when neither error analysis or chronometrics are used. In a sense, these approaches consist almost *entirely* of task analysis. While such work is typically preliminary to further work in either error analysis or computer simulation (or both), it is often useful for its own sake, as it clarifies the nature of the tasks facing children. As Kellman (1988) notes: "The realization that investigation of psychological processes presupposes a highly developed, abstract analysis of the task and available constraints has perhaps been the major advance in psychology in the last several decades" (p. 268).

Klahr and Wallace's (1972) task analysis of class inclusion is an example of such a formal characterization of an important developmental task. Their goal was to illustrate how a common "Piagetian experimental task" (i.e., the full set of components involved in the class inclusion task, including finding some objects, finding all objects, comparing

subsets of objects, etc.) involved the coordination of several more basic information processes. They proposed a network of interrelated processes (similar to Gagne's learning hierarchies) in which some processes had common subcomponents, while others were relatively independent. Klahr and Wallace's analysis enabled them to explain how surface variations in a task could invoke different processes, that, in turn, would have profound effects on performance, even though the underlying formal logic of the task remained invariant.

In the area of children's counting, Greeno, Riley and Gelman (1984) formulated a model for characterizing children's competence. Their model is much more complex than the early Klahr and Wallace analysis of classification, but it is fundamentally similar with respect to being a formal task analysis whose primary goal is to elucidate the relations among a set of underlying components. Klahr and Carver's (1988) work on debugging Logo programs provides another example of detailed task analysis. Based on their analysis of the components of the debugging process, they formulated a set of "cognitive objectives" for insertion in a programming curriculum. In addition to the instructional elements, their debugging model provided a framework for assessment of debugging skills, for creation of transfer tasks, and for evaluation of transfer.

1.1.3 Topic areas and subject populations

There is, at best, a loose association between the use of information-processing approaches and the choice of topic and/or subject population. The developmental topics studied within this approach range from higher cognitive processes, such as problem solving (Resnick and Glaser, 1976) and scientific reasoning (Kuhn and Phelps, 1982; Dunbar and Klahr, 1988), to more basic processes, such as attention and memory (Chi, 1981; Kail, 1984). Subject populations typically range from toddlers, through pre-schoolers, to late adolescents, and are typically normal, although gifted (Davidson, 1986), aging (Hoyer & Familant, 1987; Madden, 1987), and retarded and learning-disabled (Geary, et al., 1987; Spitz & Borys, 1984) populations have been studied under the information-processing rubric. In the case of special populations, issues are usually framed by the theoretical or empirical results emerging from studies of normal populations, and the question of interest is the qualitative or quantitative difference in a particular information-processing construct. For example, Spitz and Borys (1984) have studied the differences in search processes between normal and retarded adults on the classic Tower of Hanoi puzzle.

Because the focus of this chapter is *cognitive* development, I have drawn the conventional -- and arbitrary -- boundary that precludes an extensive discussion of perceptual/motor or language development. I can find no principled basis for excluding either of these areas from mainstream information processing, for in both of them one can find many examples of the approach (cf. MacWhinney, 1987; Yonas, 1988). MacWhinney's (1987) edited volume on mechanisms of language acquisition contains an array of information-processing approaches that run the gamut from soft- to hard-core features. In the area of perceptual development, Marr's (1982) seminal work, which advocates computational models as the proper approach to constructing theories of vision, is increasingly influential. Indeed, Banks (1988), in presenting his own computational model of contrast constancy, argues that perceptual development is a much *more* promising area in which to construct computational models than cognitive or social development, because there are more constraints that can be brought to bear to limit the proliferation of untested (and untestable) assumptions. Nevertheless, for reasons

of brevity, neither perceptual/motor nor language development will be treated extensively in this chapter.

1.1.4 *Soft-core information processing: What's not included?*

Even with these caveats and exclusions, the soft-core version of the term *information processing* has become so pervasive in cognitive development that it appears to have achieved the same dubious status as *structuralism*, of which Flavell (1982) says, with characteristic insight and frankness:

I think ... that we should give up using 'structuralism' and 'structuralist' to describe 'them' versus 'us' type differences of opinion about the nature and development of cognition. In my opinion, they have become empty slogans or buzz words ... They actually interfere with communication because they give one only the illusion of understanding exactly what claims about the formal aspects of development are being made. ... If someone told me [that he was a structuralist] today, I would: (1) have only a rough idea what he meant; and (2) suspect that he might also have only a very rough idea what he meant. (p. 5)

If we substitute *soft-core information processing* for *structuralism* in this quotation, Flavell's argument is equally valid. Consider the nearly universal acceptance of theoretical constructs such as short-term and long-term memory, controlled and automatic processes, encoding, storage and retrieval, schemas, frames, declarative and procedural knowledge, and so on. As Flavell summarizes his position on structuralism: "How many cognitive developmentalists can you think of who do *not* believe that the child's mental contents and processes are complexly organized? (p. 4) Similarly, who would deny that children's cognition involves the processing of information?

If this position is accepted, then the writer of a chapter on information processing has two choices: either write a comprehensive review of the state of the art in a large number of areas of cognitive development, or focus on a more limited domain -- that of hard-core information processing. The main reason not to follow the first of these two paths is that it has been done repeatedly and ably in recent years (cf. Siegler 1983, 1985; Miller, 1983; Kail & Bisanz, 1982), and it unlikely that I could improve upon those efforts. Therefore, I have chosen to follow the second path, and, for the remainder of this paper, I focus on hard-core information-processing approaches to cognitive development. I will start by describing what I mean by this term.

1.2 **Hard-core information-processing approaches to cognitive development**

The three hard-core features, shown at the bottom of Table 1, are: use of computer simulation models, non-metaphorical interpretation of such models, and creation of self-modifying systems as theories of cognitive development. These features can be viewed as the extreme points of several of the soft-core features listed in the upper portion Table 1 and described earlier. Soft-core features S1, S2, S4, and S7, have an extreme form in H1, the use of computer simulation, and H2, the interpretation of the simulation as a theoretical statement. Methodological features S5 and S6 support the evaluation of such models, and H3 is the hard-core version of S3.

1.3 H1: Use of computer simulation

Computer simulation is often viewed as *the* criterial attribute of hard-core information processing. Klahr and Wallace (1976) characterize the approach as follows:

Faced with a segment of behavior of a child performing a task, we posit the question: "What would an information-processing system require in order to exhibit the same behavior as the child?" The answer takes the form of a set of rules for processing information: a computer program. The program constitutes a model of the child performing the task. It contains explicit statements about the capacity of the system, the complexity of the processes, and the representation of information -- the data structure -- with which the child must deal.(p. 5)³

Although the resultant computer program may be sufficient to generate the same behavior as the child, there is, of course, no guarantee that every component of the program is necessary, nor that the program is unique. How then, can we gain some confidence that the program is a plausible theory?

Simon (1972) proposed four general metatheoretical constraints that can be used to evaluate computer simulation models: (a) consistency with what we know of the physiology of the nervous system; (b) consistency with what we know of behavior in tasks other than the one under consideration; (c) sufficiency to produce the behavior under consideration; and (d) definiteness and concreteness. The extent to which these constraints have been met by computer simulators varies inversely with the order in which they are listed above. Any running program satisfies the last constraint, and if it is highly task-specific, then an ingenious programmer can usually satisfy criterion c. A common criticism of this kind of simulation is the non-identifiability of the proposed model. That is, for a single task, a model is typically ad-hoc, and, in principle, an infinite number of alternative models could account for the same data.⁴ However, as we expand the range of data for which the model can account, the force of the non-identifiability criticism is weakened. For example, in the area of adult cognition, there are programs that can model behavior in a wide variety of tasks within a general category (e.g., Newell & Simon's General Problem Solver, or Feigenbaum & Simon's EPAM) and that therefore begin to satisfy constraint (b). Developmental examples are much harder to find: I can think of only one running simulation that is both constrained by a large amount of data on children's performance and applicable to a fairly disparate set of tasks (addition, multiplication, spelling, memory rehearsal), and that is Siegler's (1986) strategy choice model.

But even Siegler's work is unconstrained by the first of Simon's four criteria: the underlying physiology of the brain. Here, his model is in company with virtually all other symbolically-oriented simulations of higher order cognition, be they developmental or not.

³Interestingly, this quotation comes from a section entitled "*The Information-processing paradigm*", which contradicts my opening comments about multiple perspectives, and reveals how hard it is to keep an open mind about one's preferred approach to a field!

⁴Although such alternative models are rarely forthcoming!

For many years, computer simulators simply ignored the physiological constraint, while acknowledging that, ultimately, symbol systems were grounded in a neural substrate. This is not to say that their models were inconsistent with what was known about physiology, only that there was no consistency check at all.

However, recent analysis by Newell (1986, 1988b) of temporal constraints in cognition illustrates how the physiological constraint can be brought to bear on computer simulation models. The path is indirect: It occurs through consideration of the different hierarchical levels of the human cognitive system and time scale of operation of each level. Each level is comprised of organized assemblies of the level below it, and it runs more slowly. Newell uses very rough approximations for the operational time scale of each level: 1 ms for neurons, 10 ms for neural circuits comprised of neurons, 100 ms for a deliberate cognitive act, 1 sec for a cognitive operation. Newell (1988b) concludes:

The *real-time constraint on cognition* is that the human must produce genuine cognitive behavior in ~ 1 s, out of components that have ~ 10 ms operation times (p. 10). The significance of such a mapping, however approximate, should not be underestimated. For years, cognitive psychology has enjoyed the luxury of considering its analysis to be one that floats entirely with respect to how it might be realized in the brain ... The floating kingdom has finally been grounded. (p. 12)

How might we apply these time constraints in evaluating computer simulation models? To illustrate, I will propose a particularly far-fetched example, by considering whether or not a artificial-intelligence program, written to play high-quality chess, could be taken as a plausible theory of how humans play the game. The program, called Hitech (Berliner and Ebeling, 1988), is currently rated at a level equal to the high end of the Master level for human tournament play, so it clearly meets the criterion of being sufficient to generate the behavior of interest. Hitech gets its power by generating a massive search (about 100 million positions per move). Although there is abundant evidence that humans do not generate even a millionth as many positions, we will limit this evaluation of Hitech to temporal considerations alone, and consider only the *rate* at which Hitech generates alternative positions -- about 175,000 positions per second. Given the fact that the representation for a chess position is a complex symbol structure, requiring several elementary steps in its generation, and that the order of magnitude of neural firing rates is only about 1 ms, then the 5 *microseconds* per position rate for Hitech simply rules it out as a plausible theory of human cognition. Even if we posit a massively parallel computation (indeed, Hitech is comprised of a set of simultaneous processors), this does not make Hitech any more plausible as a human model, for, as Newell notes (chap 3), even connectionist models require time for "bringing the results of computations in one part of the network into contact with developing results in other parts of the network. (p. 91)" Both serial, symbolically-oriented computing and parallel distributed computing are constrained by the temporal requirements of aggregating results over lower levels, and the elementary processing rates -- determined by the underlying physiology of the neural tissue -- could not support a theory based on the Hitech organization.

Note that Simon's criteria for evaluating computer simulation models are similar to the set of criteria for evaluating any representation -- computer simulation or otherwise -- listed in Section 1.1.2. However, they differ in two respects. First, since they are directed toward computer simulation, Simon's criteria are stricter about actually

generating behavior and about definiteness. Second, they do not include the developmental tractability criterion listed earlier. Recall that the purpose of this criterion is to evaluate the extent to which different models of the child at two different points in time can be integrated into a transitional theory: one that can actually transform the early state into the later one. Regardless of the predictive power or elegance of a theory for a given level of knowledge, if there is no plausible mechanism that might have produced that state from some previous one, then, from the viewpoint of the developmental psychologist, such a theory is seriously deficient. Here too, Siegler's model gets good marks, because learning and performance are intertwined such that what the model does affects what it learns, and what it has learned depends on what it has done in the past.

However, if we run the clock backwards on Siegler's model, we run up against the developmentalist's equivalent of St. Augustine's musings about the "Prime Mover". Siegler's model implies that the distribution of associations between problems and responses derives from their previous distribution and environmental history. Current answers depend on answers generated in response to previous problems. But this backward induction cannot go on indefinitely, for at each of the earlier knowledge levels, we face the same question of how *that* knowledge got there. Ultimately, we come to the initial situation in which all answers have flat distributions of associations, and subjects must use fall-back strategies. But from where do those strategies, and the strategy-choice mechanism itself, originate?

Here we are forced to make assertions about what I have called the *innate kernel*. To the best of my knowledge, there are no complete proposals for what the innate information-processing system might have to contain (although Wallace, Klahr & Bluff, 1987, did outline some of the requirements). I believe that this remains one of the greatest challenges facing developmental theorists. The answer will undoubtedly require a convergence of analytic tools, such as the formulation of cognitive architectures and detailed studies of neonatal functioning. These empirical studies will be necessarily limited to the assessment of perceptual and motor behavior and will thus press the very boundaries of current approaches to information-processing psychology.

The "definiteness and concreteness" criterion is elaborated in Gregg and Simon's (1967) four main claims for the advantages of computer simulation models. The first has to do with avoidance of inconsistency: the same set of operations are used for all cases of testing the theory. While it is true that programs have an unlimited potential for points of modification, once a theory has been formulated as a program, it cannot be *inadvertently* "tuned" to special cases. My own experience in formulating several different strategies for the TOH problems (cf. Klahr & Robinson, 1981) made me appreciate how important it was to be confident that each program followed its unique rules in a consistent fashion for the 40 problems that it had to solve. The second item on Gregg and Simon's list of advantages is the elimination of implicit assumptions. Everything in a program must be stated as an unambiguous operation. Continuing the example, the creation of the alternative strategies made it very clear exactly what the differences were in each strategy, and what their implications were for performance. The third feature is unambiguous predictions: the program generates behavior that can be compared with human performance. Finally, Gregg and Simon emphasize encoding explicitness. The need to create data structures for the program to process avoids finessing questions about encoding and representation. Although one may disagree with any specific encoding, computer models require explicitness about just what goes into that encoding, and in some cases suggest further experimentation.

Neches (1982) offers a thoughtful tempering of these arguments. Although he points out that the claim for superiority of computer simulation over verbal or mathematically stated theories has sometimes been overstated, his own work on HPM -- to be described later in this paper -- actually exemplifies many of these merits. Furthermore, while it is true that the benefits listed above begin to accrue from *any* move toward formalization (as suggested by the earlier quotation from Siegler on his use of semantic nets and flow charts), the discipline of computer simulation represents a qualitative increase in all of them. Indeed, in subsequent work, Siegler's models of children's strategy choice on arithmetic tasks became sufficiently complex that the only feasible way to develop the theory and derive predictions from it was to use computer simulation (Siegler, 1986, p. 109).

There are several other instances in the developmental literature in which models initially stated in some non-computer formalism were deemed by their creators to be sufficiently imprecise, complex, or ambiguous to require further specification as computer simulations: Shultz's (1987, 1988) models of causality, Halford's work on structure-mapping (Bakker and Halford, 1988), and Gentner's research on analogy and metaphor (Gentner, 1988; Falkenhainer, Forbus, and Gentner, 1986) all exhibit this tendency to move to computer simulation as theory development matures.

1.3.1 *The computer's role in simulation models*

Given the centrality of computer simulation to hard-core information processing, it may be useful to clarify a few essential points that are often misunderstood. First of all, it is important 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 usually sufficiently complex that it is necessary to run them on a computer in order to explore their implications, but this does not imply that the theory bears any resemblance to the computer on which it runs. Computer simulations of hurricanes do not imply that meteorologists believe that the atmosphere works like a computer. Furthermore, the same theory could be implemented on computers having radically different underlying architectures and mechanisms.

Failure to make this distinction leads to the common misconception that information-processing approaches can be arranged along a dimension of "how seriously they take the computer as a model" (Miller, 1983). It would be counterproductive for a developmental psychologist to take the computer at all seriously as a model for cognition, because the underlying computer does *not* undergo the crucial self-modification necessary for cognitive development. A similar misunderstanding of the role of the computer in hard-core information-processing models may have led to Brown's (1982) widely quoted (but misdirected) criticism 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." I agree: later in this chapter, I will describe some hard-core information-processing approaches that propose very explicit mechanisms for "adaptive modification to a changing environment." The hard-core information-processing approaches are serious, not about the similarity between humans and computers, but rather about the extent to which intelligent behavior -- and its development -- can be accounted for by a symbol-processing device that is manifested in the physical world. The strong postulate for hard-core information-processing is that both computers and humans are members of the class of "physical symbol systems" (Newell, 1980), and that some of the theoretical

constructs and insights that have come out of computer science are relevant for cognitive developmental theory.

1.3.2 Recursive decomposition and emergent properties

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 hard-core information processing: "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, pp. 52-52)

Palmer and Kimchi illustrate this point with the memory-scanning process described earlier: it is accomplished by the appropriate organization of simpler processes: matching one symbol to another, moving through an ordered list, setting an indicator for whether the probe has been matched or not, etc. None of these sub-processes, in isolation, does a memory scan. Indeed, each of them could be used in quite a different super-process, such as sorting a list. It is their *organization* that gives them the emergent property of being a scanning process.

The importance of emergent properties cannot be overemphasized, for it provides the only 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 "basic processes" at a much higher level -- be it production systems or networks of activated nodes, emergent properties continue to emerge, for that is the nature of complex systems.⁵ Siegler's model of children's strategy choice in arithmetic provides an interesting developmental example of the emergent property of a rational choice of an efficient and effective strategy. In that model, strategy choices about whether to retrieve the answer to a multiplication problem from memory or to calculate the result are made without any rational calculation of the advantages and disadvantages of each strategy. As Siegler (1988) puts it: "Rather than metacognition regulating cognition, cognitive representations and processes are assumed to be organized in such a way that they yield adaptive strategy choices without any

⁵There is, at present, a vigorous debate taking place within cognitive science as to the appropriate level at which to represent the primitive, non-decomposable components, and how to account for their organization. The "symbol-processors" tend to start with the symbol, and to construct intelligence out of symbolic structures, while the "connectionists" (Rumelhart and McClelland, 1986) start with distributed patterns of activation over networks of nodes. I will not go into that debate in this paper, but note that in both cases there is fundamental agreement that intelligence is an emergent property based on the organization of components. The intelligence derives largely from the architecture.

direct governmental process." Although this may sound like Adam Smith's "invisible hand" applied to the mental marketplace, it exemplifies the idea of emergent properties in the context of an interesting developmental phenomenon.

The emergent property notion provides the key to my belief that hard-core information processing has the potential to formulate 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. As I noted earlier, we do not yet have a detailed proposal for what the innate kernel is, and, with respect to the ambitious goal of creating a full account of the development of the information-processing system, Siegler's example may seem like a small step, but it is a step in the right direction. I will describe a few others below.

1.3.3 Data constraints

Another aspect of simulation models that tends to be misunderstood is the extent to which they can be said to account for data. For example, Liben (1987) claims that using simulation models to account for empirical results is circular because:

...the competence model is empirically derived directly from observed performance, as illustrated in [work by Siegler and Shipley (1987)]. That is, given that the particular computer program was written expressly to simulate children's observed behaviors, it is not remarkable that there is a good match between them." p. 114)

Beilin (1987), echoing Liben, asserts that:

Inasmuch as computer simulations usually mimic the data and performances they are designed to predict, such predictions usually turn out to be successful.

It is hard to make sense of these simplistic criticisms as they stand. A minor paraphrase of Liben reveals why:

Given that Newton's inverse-square law of gravitation was formulated expressly to account for empirical observations of planetary motion, it is not remarkable that there is a good match between his theory and the actual motion of the planets.

The problem is that unless one understands *how* a theory generates its predictions, it is impossible to assess its circularity or remarkability. This is true no matter what the form of the theory: be it a computer simulation, a mathematical model, or a verbal statement. In the case of a computer model, one could generate a perfect fit to subjects' behavior by simply reading in a data table, derived from subject performance, and then printing it out again --- hardly an interesting exercise. On the other hand, if the model is based on a set of basic processes and a few parameters, and if, furthermore, the model makes testable predictions about data patterns that were not detected before the model was formulated, then it serves the role that any theory should. That is, it summarizes existing data patterns and predicts new ones on the basis of fundamental principles.

The additional advantage of computer simulation models over conventional forms of theorizing is that they permit a very clear allocation of "credit" for such fits and predictions to the various sources: the general theoretical principles, the particular parameter values in the model (one can explore the parameter space in a model to discover just which variables are critical, and to which ones the model is relatively insensitive), or the particular encoding of the task environment. In contrast, with verbal models or even flow charts, it is never clear how much of the interpretive work is being done by the theory, and how much by the reader of the theory.

1.4 H2: Commitment to elements of the simulation as theoretical assertions, rather than just metaphor or computational convenience

This is another aspect of Miller's (1983) question about how seriously one should take the computer as a model of thought. The degrees of seriousness here are not about the correspondence between the computer and the theory, but about the *program* and the theory. In some cases, the program is used as a convenient format for stating a set of processes that could be implemented in many equivalent forms and in many computer languages. The program's role is to compute the behavior of the system under a set of specified inputs. Klahr and Robinson's (1981) simulation models of children's performance on the Tower of Hanoi puzzle exemplify this soft end of the computer-simulation attribute.

At the other end of the attribute, the program and the computational architecture that interprets it (i.e. runs the program) jointly comprise a theoretical statement about the general organization of the cognitive system and the specific knowledge that is required to do the task at hand. Perhaps the most commonly proposed architecture for this kind of hard-core model is a production system. Both the production-system interpreter and the specific productions are proposed as theoretical constructs, not just programming conveniences. For example, Klahr and Wallace (1976) utilized Newell's original production system architecture⁶ to formulate a theory of the development of quantitative processes including elementary quantification, class-inclusion, transitive reasoning, and conservation of quantity. Programs for all of these tasks were constrained by the theoretical principles embodied in the production-system architecture, and the entire package was intended to be "taken seriously."

In production-system models, the productions and the architecture bear the same relation to cognitive behavior as a particular molecular structure and general laws of chemistry are taken to jointly explain the behavior of a substance. For the hard-core simulator using productions, it is no more appropriate to argue that productions are only functionally equivalent to some "real" mental item, than it is to say that molecules are

⁶Although the use of production systems to model human performance had been introduced several years earlier by Newell (1968), it wasn't until the early 70's that Newell produced a running system for general use (Newell, 1973; Newell & McDermott, 1975). This turned out to have a profound impact in two related domains: within cognitive psychology, it was the first serious proposal for a "tool kit" for building simulation models based on a well-defined cognitive architecture. (The expansion of Newell's original production-system architecture into a large space of such architectures will be discussed in the next section.) Within artificial intelligence, production systems spawned an industry dedicated to the creation of computer-based expert systems. (See Neches, Langley & Klahr, 1987, for a brief history of production systems in psychology, and Brownston, Farrell, Kant and Martin (1985) for a tutorial on building expert systems.)

only functionally equivalent to some real chemical entity. The production-system architecture is sufficiently important to hard-core information processing that I will describe it at length in the Section 2.

1.5 H3: Goal of creating a complete self-modifying simulation that accounts for both task performance and development

The objective.

- Specify an innate kernel -- cast as a self-modifying production system -- that characterizes the neonate information-processing system.
- Represent the external environment in such a way that the system can utilize its perceptual and motor operators to interact with the environment, and to learn from that interaction.
- Run the system and let it develop its own intelligence.

That is the Holy Grail of the hard-core information-processing approach to cognitive development. The question is whether the enterprise is under the control of Tennyson or Monty Python. My own bets are with the Idylls of the King, as I believe that self-modifying production systems are able to represent and account for the fundamental inseparability of performance and change. Some important pieces of this puzzle are already in place, but much remains to be accomplished before we will have the kind of total system envisioned above. In the following section, I will lay out some of the major issues in the use of production systems that must be resolved in order to achieve the ultimate goal.

2 Production systems: At the core of the core⁷

Production systems are a class of computer-simulation models stated in terms of condition-action rules. A production system consists of two interacting data structures, connected through a simple processing cycle:

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 is comprised of three distinct processes:

1. The *match* process finds productions whose conditions match against the current state of working memory. The same rule may match against working

⁷Parts of this section have been adapted from Neches, Langley & Kuhr, 1987

memory in different ways, and each such mapping is called an *instantiation*. When a particular production is instantiated, we say that its conditions have been *satisfied*. In addition to the possibility of a single production being satisfied by several distinct instantiations, several different productions may be satisfied at once. Both of these situations lead to *conflict*.

2. The *conflict resolution* process selects one or more of the instantiated productions for applications.
3. The *act* process applies the instantiated actions of the selected rules, thus modifying the contents of working memory.

The basic recognize-act process operates in cycles, with one or more rules being selected and applied, the new contents of memory leading another set of rules to be applied, and so forth. This cycling continues until no rules are matched or until an explicit halt command is encountered. The many variations that are possible within this basic framework will be described in Section 2.3.

2.1 Notation or theory?

The distinction made earlier between two related interpretations of the theoretical status of computer simulation models applies to production-system models. Under the first interpretation (feature S5), production systems are simply a formal notation for expressing models, and the object of interest is *model content*, rather than expressive form or interpretation scheme. For example, one might characterize the rules a person uses to perform some task in terms of a production system without necessarily committing to the psychological assumptions inherent in the production system interpreter. Other formalisms for expressing the same content are possible (e.g., scripts, LISP programs, and flowcharts), and one can debate their relative merits (see Klahr and Siegler, 1978).

In contrast, the hard-core view (feature H3) treats both the task-specific productions and the production-system interpreter as theoretical assertions about domain-dependent and domain-independent components of behavior. That is, the production system interpreter serves as a particular theory about the architecture of the human information processing system. This view was originally put forward by Newell (1967, 1972) and substantially extended by Anderson (1983). Most recently, it has been reformulated as a major theoretical statement by Newell (1988a). He asserts that humans actually employ productions in language, reasoning, motor skill, and every other form of intelligent behavior, and he describes a novel form of production system architecture -- called SOAR -- that is proposed as a unified theory of human cognition.

The developmental relevance of this hard-core view derives from the ability of production system models to modify themselves in ways that capture many of the central features of learning and development. This potential for self-modification provides the major justification for the use of production systems in modeling cognitive development. In the following sections, I summarize some issues surrounding the adoption of production systems as a candidate for the cognitive architecture of the developing human.

2.2 Properties of production-system models

Newell and Simon (1972) summarized the production system features that recommend them for modeling human behavior as follows:

1. *Homogeneity.* Production systems represent knowledge in a very homogeneous format, with each rule having the same basic structure and carrying approximately the same amount of information.
2. *Independence.* Productions are independent of one another in the sense that one production makes no direct reference to any other production. Their interaction occurs only through their effects on working memory. Therefore it is easy to insert new rules or remove old ones. This makes production systems a very congenial format for modeling successive stages in a developmental sequence and also makes them attractive for modeling the incremental nature of much human learning.
3. *Parallel/serial nature.* Production systems combine the notion of a parallel recognition process with a serial application process; both features seem to be characteristic of human cognition.
4. *Stimulus-response flavor.* Production systems inherit many of the benefits of stimulus-response theory but few of the limitations, since the notions of stimuli and responses have been extended to include internal symbol structures.
5. *Goal-driven behavior.* Production systems can also be used to model the goal-driven character of much human behavior. However, such behavior need not be rigidly enforced; new information from the environment can interrupt processing of the current goal.
6. *Modeling memory.* The production-system framework offers a viable model of long-term memory and its relation to short-term memory, since the matching and conflict resolution process embody principles of retrieval and focus of attention.

2.3 Production systems as cognitive architectures

As noted earlier, the term "cognitive architecture" denotes the invariant features of the human information processing system. Since one of the major goals of any science is to uncover invariants, the search for the human cognitive architecture should be a central concern of developmental psychology. The decision to pursue production system models involves making significant assumptions about the nature of this architecture. However, even if one accepts a production-system framework for formulating developmental theories, many decisions remain to be made. Theory formulation takes on the properties of a constrained design process. There is a general framework, within which particular architectural design options must be further specified. Once made, the resultant production-system interpreter represents one point in a large design space. That is, it is a specific theory of the human cognitive architecture, within the general production-system framework. The evaluation of the theory then rests on the kinds of criteria listed earlier. At present, there are no proposals for a complete developmental

architecture of this type, but there are some candidates for the adult system that could be extended to play this role. Later in this chapter, I will briefly describe one such architecture.

Before getting to that, I will lay out the major dimensions of the space of production-system architectures. Within the general framework, production system interpreters can differ along four major dimensions: working memory management, the structure of production memory, conflict resolution policies, and self-modification mechanisms. I will discuss the first three of these briefly, and then in Section 2.5 elaborate the self-modification issue.

2.3.1 Working memory issues

1. *The structure of memory.* Is there a single general working memory, or multiple specialized memories (e.g., data and goal memories, or memories for interface with the perceptual and motor environments)? In the latter case, how are conditions in productions specialized to match particular memories?
2. *The structure of elements.* What is the basic form of working memory elements (e.g., list structures, attribute-value pairs)? Do elements have associated numeric parameters, such as activation or recency?
3. *Decay and forgetting.* Are there limits on the number of items present in working memory? If so, are these time-based or space-based limitations?
4. *Retrieval processes.* Once they have been "forgotten," can elements be retrieved at some later date? If so, what processes lead to such retrieval? For example, must productions add them to memory, or does "spreading activation" occur?

2.3.2 Production memory issues

1. *The structure of memory.* Is there a single general production memory, or are there many specialized memories? In the latter case, are all memories at the same level, or are they organized hierarchically?
2. *The structure of productions.* Do productions have associated numeric parameters (e.g., strength and recency) or other information beyond conditions and actions?
3. *Expressive power of conditions.* What types of conditions can be used to determine whether a rule is applicable? For example, can arbitrary predicates be included? Can sets or sequences be matched against? Can many-to-one mappings occur?
4. *Expressive power of actions.* What kind of processing can be performed within the action side of an individual rule? For example, can arbitrary functions be evoked? Can conditional expressions occur?
5. *Nature of the match process.* Are exact matches required or is partial

matching allowed? Does the matcher find all matched rules, or only some of them? Does the matcher find all instantiations of a given production?

2.3.3 Conflict resolution issues

1. *Ordering strategies.* How does the architecture order instantiations of productions? For example, does it use the recency of matched elements or the specificity of the matched rules?
2. *Selection strategies.* How does the architecture select instantiations based on this ordering? For example, does it select the best instantiation, or does it select all those above a certain threshold?
3. *Refraction strategies.* Does the architecture remove some instantiations permanently? For example, it may remove all instantiations that applied on the last cycle, or all instantiations currently in the conflict set.

To summarize, the basic production-system framework has many possible incarnations, each with different implications about the nature of human cognition. Of particular importance to cognitive development are the self-modification issues, but before turning to a more extensive discussion of them, I will briefly describe some *non-self-modifying* production-system models of children's performance in a few domains of importance to developmental psychology.

2.4 Some examples of production-system models of children's performance

Even when cast as models of different performance levels, rather than as models of transition processes, production-system simulations can serve useful functions. In this section I describe four different ways -- taken from my own research -- in which *non-self-modifying* production systems have been used to model children's performance. The first example illustrates how production systems can be matched to chronometric data to produce some estimates of the duration of elementary components of the recognize-act cycle. The second example illustrates one of the most valuable features of production systems for modelling cognitive development: the ease with which different performance levels can be represented by a family of models having different production sets. The third example focuses on how production systems can include encoding and performance productions in the same general format, and the final example illustrates a kind of "vertical integration" in a production-system model that represents several levels of knowledge from general principles down to specific encoding rules.

2.4.1 Quantification: Matching production firings to chronometric data

Production-system models of thinking were initially developed to account for the verbal protocols generated by subjects working on puzzles requiring several minutes to solve (Newell, 1966). However, a much finer temporal grain of analysis was used in the first production-system models that actually ran as computer simulations. Newell (1973) introduced his production-system language (PSG)⁸ in the context of the Sternberg

⁸An acronym for "Production System G". This implies that six precursor versions had already been deemed unsuitable for public consumption; I take this as an indirect testimony to Newell's standards of excellence.

memory-scanning paradigm (described in Section 1.1.2) In the same volume (Chase, 1973), I described a model, written in PSG, of elementary processes for quantification: subitizing, counting, and adding (Klahr, 1973). Both of these models were atypical of most subsequent production-system models in that they attempted to account for chronometric data in terms of the dynamic properties of the production-system execution cycle. That is, they estimated the duration of specific micro-processes within the recognize-act cycle (such as the time to do a match, or the time to execute an action) by relating the number of such micro-process executions to the reaction-time data.

Although neither of these early models dealt with developmental data, the model of elementary quantification processes was subsequently elaborated into one that did deal with the differences in subitizing rates between children and adults (Klahr & Wallace, 1976 Chaps 3 and 8). The elaboration included two distinct "working memories": one corresponding to the traditional STM, and the other corresponding to an iconic store. Accordingly, the condition elements in productions could refer to either of these information sources, and the time parameters associated with matches in the two stores differed.

By attempting to constrain the model-building process with the chronometric data from very different domains, both of these models converged on a gross estimate of the time duration for the basic production-system cycle time of between 10 and 100 ms. While this may seem to be a fairly loose parameter estimate, it is important to note that it is not 1 ms, nor is it 1000 ms. That is, if the production cycle is constrained, even within these broad limits, then one can evaluate the plausibility of particular production systems in terms of whether they exhibit -- within an order of magnitude -- the same *absolute* as well as relative temporal patterns as do the humans they are modelling.

2.4.2 Production systems for different levels of performance

In contrast to relatively rare chronometrically-constrained production systems, the "family of models" approach is the most common use of production systems by developmentalists. The goal here is to produce a family of production-system models for a specific task that represent different levels of performance. Once it has been demonstrated that the models can indeed produce the appropriate behavior at each level of performance, then one can examine the differences between successive models in order to infer what a transition mechanism would have to accomplish. Baylor and Gascon (1974) did this kind of analysis for levels of weight seriation, and Klahr and Siegler (1978) did it for the balance scale task. Siegler previously had produced an elegant analysis of rule sequences characterizing how children make predictions in several domains (Siegler, 1976), and the sequences were formulated as a series of increasingly elaborated binary decision trees. 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 just the decision tree representation. Even without describing the models, the following quotation from their paper conveys the level of detail that was facilitated by the production-system formulation.

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

modification changes the action side from random muddling through to "get torques.")

We can 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. 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. 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 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. Model IV 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 (p. 80).

2.4.3 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 some models to account for some children's idiosyncratic -- but consistent -- response patterns. One 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 a lot of other knowledge about the task context:

The model represents, in addition to the child's knowledge about how the balance scale operates, her knowledge about the immediate experimental context in which she is functioning. The trial-by-trial cycle during the training phase comprises (1) observation of the static display, (2) prediction of the outcome, (3) observation of the outcome, (4) comparison of the outcome with the prediction, and (5) revision if necessary of the criterion. This model utilizes, in one way or another, representation of knowledge about when and how to encode the environment, 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. (Klahr & Siegler, 1978, p. 89)

This kind of model raises two issues that might otherwise escape notice. First, what

kinds of knowledge are necessary to generate these different encodings, and where do they come from? It has long been known that "surface" variations in tasks can cause wide variation in children's performance -- even on the tasks purported to index developmental level, such as class inclusion (Klahr and Wallace, 1972). Production-system formulations avoid the arbitrary dichotomy between "performance" demands and the so-called "logical" properties of a task, and force an unambiguous specification of all the processing necessary to complete the task. Second, how much of the encoded knowledge (i.e., the contents of working memory) must be available at any one moment? That is, in order to do the task, how much working memory capacity is required? Case (1986) addresses this issue informally in his proposed procedures for quantifying tasks in terms of their demands on the Short Term Storage Space. However, without a clear and principled specification of the grain-size and computational power of the routines that use the contents of STSS, it is difficult to apply his demand-estimating procedure to a new domain.

2.4.4 Multiple-level production system: From principles to encodings

Klahr and Wallace (1976) describe a model of children's performance on Piaget's conservation of quantity task. Their model contains productions dealing with several different levels of knowledge. At the highest level are productions that represent general conservation principles, such as "If you know about an initial quantitative relation, and a transformation, then you know something about the resultant quantitative relation." (See Klahr and Wallace, 1973, for an elucidation of these conservation principles.) 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. These are the same productions that comprised the systems described earlier in our discussion about matching production systems to chronometric data.

Although I have described this system as if there were a hierarchy of productions, there is only the flat structure of a collection of productions. Each production simply checks for its conditions. If it fires, then it deposits its results in working memory. The hierarchy emerges from the specific condition elements in each production, which ensure that productions only fire when the current context is relevant.

2.4.5 Non-transition models: A summary

Recall that in preparation for this recent enumeration of computer simulations of developmentally-relevant phenomena, I first limited the discussion to production systems, then to state models, rather than transition models, and finally, for convenience, to the work I know best. As a result, I have traversed a familiar, but narrow, path. However, these four instances by no means exhaust the set of computer simulations of children's thinking processes. Rabinowitz, Grant and Dingley (1987) summarize over a score of other computer simulation models relevant to cognitive development, including those that use non-production-system architectures, and including both state and transition models. The production-system models include work on seriation (Baylor, Gascon, Lemoyne, and Pother, 1973; Young, 1976) and subtraction (Young and O'Shea, 1981). Computer simulations based on schema architectures have been proposed in the area of arithmetic

(Greeno, Riley and Gelman, 1984; Riley, Greeno and Heller, 1983; Kintsch and Greeno, 1985) and language acquisition (Hill, 1983). Task-specific architectures have been used to model children's performance on addition (Ashcraft, 1987; Siegler, 1988), subtraction (Brown and VanLehn, 1982), and series completion (Klahr and Wallace, 1970b).

As Rabinowitz et al. note, only a handful of these models include any self-modifying mechanisms. Nevertheless, the underlying assumption in all of the computer simulations is that by clarifying the nature of children's thought at any particular level of development, the requirements of a transition theory become better defined. Thus, regardless of their intrinsic merits, the principle value of all of these state models is that they provide promissory notes for a model of self-modification. Furthermore, I believe that production system architectures are both highly plausible and very tractable architectures within which to formulate theories of self-modification. In the next section, I consider this issue in detail.

2.5 Self-modification

Self-modification can lay claim to being the central issue for a cognitive developmentalist. One way to approach self-modification from a production-system perspective is to assume the stance 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.

First, a definition. Rather than get side-tracked by attempting to distinguish between learning and development, I will use the more neutral term *change*, and it will be understood that the change is imposed by the system's own information-processing mechanisms (hence "self-modification"). Note that while learning is usually defined -- in one form or another -- as "the improvement of performance over time", such directionality is not necessarily implied by change. Indeed, in many areas of development, the measured trajectory is U-shaped, rather than monotone (Strauss, 1982), and a theory of change must account for this. So for now, I will use change as the generic term for self-modification, and later I will return to the question of whether self-modifying production systems are models of learning or development.

2.5.1 Mechanisms

Many general *principles* for change have been proposed in the developmental literature. These include things like equilibration, encoding, efficiency, redundancy elimination, search reduction, self-regulation, consistency detection, and so on. However, they are not mechanisms. Once we have adopted a production system architecture, we can pose the following focused questions about how these principles might be implemented as specific mechanisms.

1. *Change mechanisms.* What are the basic change mechanisms that lead to new productions? Examples are generalization, discrimination, composition, proceduralization, and strengthening.
2. *Conditions for change.* What are the conditions under which these change mechanisms are evoked: when an error is noted, when a rule is applied, when a goal is achieved, or when a pattern is detected?

3. *Interactions among mechanisms.* Do the change mechanisms complement each other, or do they compete for control of behavior? For example, generalization and discrimination move in opposite directions through the space of conditions.

The recognize-act cycle offers three points at which change can have an effect: a production system's repertoire of behaviors can be changed by affecting the outcome of (1) production matching, (2) conflict resolution, and (3) production application.

2.5.2 Change during the match

The most commonly used technique for altering the set of applicable productions found by the matching process is to add new productions to the set. As long as matching is exhaustive, the new productions are guaranteed to be considered during the next recognize-act cycle. One way to generate the new productions is to modify the conditions of existing rules. Anderson, Kline, and Beasley (1978) were the first to modify production system models of human learning via *generalization* and *discrimination*. The first mechanism creates a new rule (or modifies an existing one) so that it is *more* general than an existing rule, meanwhile retaining the same actions. The second mechanism -- discrimination -- creates a new rule (or modifies an existing one) so that it is *less* general than an existing rule, 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.

Within production-system models there are three basic ways to form more general or specific rules, each corresponding to a different view of generality. First, one can add or delete conditions from the left-hand side of a production. The former generates a more specific rule, since it will match in fewer situations, while the latter gives a more general rule. The second method involves replacing variables with constant terms, or vice versa. Changing variables to constants reduces generality, whereas changing constants to variables increases generality. The final method revolves around class hierarchies. For example, one may know that both dogs and cats are mammals and that both mammals and birds are vertebrates. Replacing a term from this hierarchy with one below it in the hierarchy decreases generality, while the inverse operation increases generality.

These techniques have been used in programs modeling behavior on concept acquisition (Anderson and Kline, 1979), language comprehension and production at various age levels (Langley, 1982; Anderson, 1981), geometry theorem proving (Anderson, Greeno, Kline, & Neves, 1981), and various puzzle-solving tasks (Langley, 1982). Note that both methods require instances that have been clustered into some class, and both attempt to generate some general description of those classes based on the observed instances. These mechanisms are described in considerable detail by Langley (1987).

2.5.3 Change during conflict resolution

Once a set of matching rule instantiations has been found, a production-system architecture still must make some determination about which instantiation(s) in that set will be executed. Thus, conflict resolution offers another decision point in the recognize-act cycle where the behavior of the system can be affected. This turns out to be particularly important because many models of human learning attempt to model its

incremental nature, assuming that learning involves the construction of successively closer approximations to correct knowledge over a series of experiences.

The knowledge represented in a new production is essentially an hypothesis about the correctness of that production. A self-modifying system must maintain a balance between the need for feedback obtained by trying new productions and the need for stable performance obtained by relying on those productions that have proven themselves successful. This means that the system must distinguish between rule *applicability* and rule *desirability*, and be able to alter its selections as it discovers more about desirability. Production systems have embodied a number of schemes for performing conflict resolution, ranging from simple fixed orderings on the rules in PSG (Newell and McDermott, 1975) and PAS (Waterman, 1975), to various forms of weights or strengths (Anderson, 1976; Langley, 1987), to complex schemes that are not uniform across the entire set of productions as in HPM (Neches, 1987), to no resolution at all, as in SOAR (Newell, 1988b).

2.5.4 Changing conditions and actions

Various change mechanisms have been proposed that lead to rules 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 rules into a new rule with the conditions and actions of the component rules. However, conditions that are guaranteed to be met by one of the actions are not included. For instance, composition of rules ($AB \rightarrow CD$) and ($DE \rightarrow F$), would produce the rule ($ABE \rightarrow CDF$). Of course, the process is not quite this simple; most composition methods are based on instantiations of productions rather than the rules themselves, and one must take variable bindings into account in generating the new rule. Lewis (1987) discusses the situations under which such compositions are likely to have the desired effects.

Another mechanism for creating new rules is *proceduralization* (Neves and Anderson, 1981). This involves constructing a very specific version of some general rule, based on some instantiation of the rule that has been applied. This method can be viewed as a form of discrimination learning because it generates more specific variants of an existing rule. However, the conditions for application tend to be quite different, and the use to which these methods have been put have quite different flavors. For instance, discrimination has been used almost entirely to account for reducing search or eliminating errors, whereas proceduralization has been used to account for speedup effects and automatization.

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-law 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 basic 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") represents the most fully-elaborated candidate for complete cognitive theory -- a "unified theory of cognition" (Newell, 1988a) -- and to give even a brief overview of SOAR would require a substantial extension of the present chapter. I will comment on only its approach to self-modification. 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 sub-goaling, and in all problem-spaces. (SOAR operates entirely through search in problem spaces: spaces for encoding the environment, for applying operators, for selecting operators, etc.) Chunking reduces processing by extending the knowledge base of the system.

2.5.5 Are other mechanisms necessary?

Langley, Neches, Neves, and Anzai (1980) have argued that self-modifying systems must address two related problems: including *correct* rules for when to perform the various actions available to the system and developing *interesting* new actions to perform. However, most of the models that have been developed in recent years have focused on the first of these issues, and some researchers (e.g., Anderson 1983) have asserted that mechanisms such as composition, generalization, and discrimination are sufficient to account for all change.

Nevertheless, it appears that although these processes may be necessary components of a computational change theory, they may not be sufficient. The evidence for this comes from a number of studies that have tried to characterize differences between the strategies employed by experts and novices (Hunter, 1968; Larkin, 1981; Lewis, 1981; Simon and Simon, 1978). The reorganization necessary to get from novice to expert level involves much more than refinements in the rules governing when suboperations are performed. Such refinements could presumably be produced by generalization and discrimination mechanisms. However, producing this new procedure requires the introduction of new operations (or at least new goal structures), such as those involved in keeping a running total of the subproducts. Those new operations, and the control structure governing the sequence of their execution, require the introduction of novel elements or goals -- something that generalization, discrimination, and composition are clearly not able to do.

There are only a few studies in which change sequences, and the intermediate procedures produced within them, have been directly observed. Fortunately, a similar picture emerges from both studies. Anzai and Simon (1979) examined a subject solving and re-solving a five-disk Tower of Hanoi puzzle. They found a number of changes in procedure that seemed inconsistent with strict composition/generalization/discrimination models. These included eliminating moves that produced returns to previously visited problem states, establishing subgoals to perform actions that eliminated barriers to desired actions, and transforming partially specified goals (e.g., moving a disk off a peg) into fully specified goals (e.g., moving the disk from the peg to a specific other peg).

In the second study, Neches (1981) traced procedure development in the command sequences issued by an expert user of a computer graphics editing system. In doing this, he found a number of changes that involved reordering operations and replanning procedure segments on the basis of efficiency considerations. Subjects were able to evaluate their own efficiency at accomplishing goals and to invent new procedures to reach the same goals more efficiently.

The important point in both of these examples is that the change appears to involve reasoning on the basis of knowledge about the structure of procedures in general, and the semantics of a given procedure in particular. In each example, procedures were modified through the construction of novel elements rather than through simple deletions, additions, or combinations of existing elements.

2.5.6 Heuristic procedure modification

This class of self-initiated qualitative improvements is exemplified by children's acquisition of the min strategy for simple addition problems discussed earlier. When children are first instructed in addition, they are taught the "count all" algorithm, but they eventually develop the min strategy on their own. Their answers are correct under execution of either strategy (but not equally -- see Siegler, 1987 for a careful analysis of the relation between errors and strategy choice) and there is no explicit instruction that tells children to *create* a min strategy. What kind of self-modification mechanism could account for this and other examples of the ubiquitous tendency for children to develop novel approaches to problems? Neches (1981, 1987) proposed a production-system architecture called HPM (for Heuristic Procedure Modification) that addresses these issues. The model demonstrates how a system can learn entirely from its own performance without relying on external feedback. From an architectural perspective, HPM's most important features are a goal trace, which leaves a record of goal accomplishments, and a production trace, which preserves information about the temporal order of production firing, and the context in which they fired.

The general idea that change systems should be able to observe their own performance appears under several rubrics, and it remains to be seen just how much they differ. HPM is one clear instantiation of the notion, and it also appears as the "time line" notion in the developmental model sketched by Wallace, Klahr & Bluff (1987). It is also captured to some extent in the way that SOAR forms chunks out of the goal trace and local context for satisfied sub-goals.

2.6 Summary: Production systems as frameworks for cognitive developmental theory

In this section I have provided both a brief overview of production-system architectures and a perspective on the issues that arise in applying them to the areas of learning and development. The framework rests on three fundamental premises of the hard-hard-core approach:

1. *The structure of production-system architectures provides insight into the nature of the human information-processing 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 relationship 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 the 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 it 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 people have finally started 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 linking.

3. *All information-processing-system architectures, whether human or artificial, must obey certain constraints in order to facilitate the process of change.* It is these constraints that give rise to the seemingly complex particulars of individual production system architectures. Thus, following from our second premise, an understanding of production-system models of change is a step toward understanding the nature of human development and learning.

I have tried to demonstrate how computer-simulation models in general, and production-system models in particular, enable us to sharpen and focus the question of self-modification in a way that is simply unattainable in more traditional verbal formulations of theories of state or transition. The early critics of information-processing models in cognitive development (Beilin, 1983; Brown, 1982) faulted these models for their lack of attention to issues of transition and change. However, they failed to understand the principal virtue of the early simulation models of distinct states: that they explicated many of the complex requirements for a self-modifying system (an explication entirely absent from Geneva accounts of equilibration). However, both the Rabinowitz et al. review and the listing in this section clearly indicate, several examples of self-modifying systems have been created and described in the literature. Nevertheless, echos of the "non-modifiability" theme are still appearing (cf. Liben, 1987, p. 117, citing Beilin, 1983), even though the existence of self-modifying systems in specific domains provides concrete evidence that the criticism is uninformed and unfounded.

3 Conclusion and speculation

In this chapter I have attempted to define and illustrate the major attributes of information-processing approaches to cognitive development. For rhetorical purposes, I proposed a dichotomy between soft-core and hard-core attributes, when in reality, they form several continua having complex and subtle interactions. The main point to be made was that at the soft end of these attributes, information-processing approaches are so pervasive as to be redundant modifiers of "cognitive development". Distinctive features only begin to appear as we approach the hard-core instances, particularly those that use computer simulation as a form of theory building. I then went on to describe the relevance and potential of a particular, theory-laden type of computer simulation: production systems. Several examples of how production systems have been used to model performance on developmentally important tasks were presented, and then I introduced self-modifying production systems and their potential for modelling change. In this final section, I will make a few general comments on the state of theorizing about developmental mechanisms, point to one area of great potential importance that has not been treated in the chapter, and speculate about the future of information-processing approaches to cognitive development.

3.1 Is this trip necessary?

Are computational models worth the effort? Why should someone interested in theories of cognitive development be concerned about the detailed architectural variations of the sort discussed earlier? The primary justification for focusing on such systems is my earlier claim that self-modification is *the* central question for cognitive developmental theory. My personal belief is that if we want to make theoretical advances, then we have no other viable alternatives than to formulate computational models at least as complex as the systems described here.

Some people have criticized the area for being insufficiently attentive to the issue of self-modification.

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, 1988a, p. 333)

Is this too harsh a judgment? Perhaps we can dismiss it as based on hearsay: for Newell himself is not a developmental psychologist. But it is harder to dismiss the following assessment from John Flavell (1984)

... 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 (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 (Banks, 1987):

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. (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, but I believe that it is no longer being avoided: many developmentalists have been at least asking the right questions recently. In the past few years we have seen Sternberg's (1984) edited volume *Mechanisms of Cognitive Development*, MacWhinney's (1987) edited volume *Mechanisms of Language Acquisition*, and Siegler's (1989) *Annual Review* chapter devoted to transition mechanisms. So the question is being asked. Furthermore, the trend is in the direction of hardening the core. Only a few of the chapters in the 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, in characterizing several general categories for transition mechanisms (neural mechanisms, associative competition, encoding, analogy,

and strategy choice) is able to point to computationally-based exemplars for all but the neural mechanisms (e.g., Bakker & Halford, 1988; Falkenhainer, et al. 86; Holland, 1986; MacWhinney, in press; Rumelhart & McClelland, 1986; Siegler, in press).

To reiterate, as Flavell and Wohlwill (1969) noted 20 years ago: "Simple models will just not do for developmental psychology". A serious theory of cognitive development is going to be enormously complex. The formulation, adaptation, or extension of a universal theory of cognition of the scope of something like SOAR is a major intellectual commitment.

A clear advantage of computational models is that they force difficult questions into the foreground, where they cannot be sidetracked by the wealth of detailed but unconnected experimental results, nor obscured by vague generalizations and characterizations about 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.

3.2 Connectionism and cognitive development

Earlier in this chapter, I justified the exclusion of information-processing models of perceptual/motor development on conventional grounds. The implication was that it was simply a matter of space constraints. However there is a more critical interpretation of the exclusion of motor and perceptual areas from the core of information-processing approaches. This view argues that information-processing approaches of the symbolic variety are inherently inadequate to account for the important phenomena in perception and motor behavior. The gist of the argument is that, given the highly parallel and "presymbolic" nature of these areas, and given the serial and symbolic nature of most information-processing accounts of higher cognition, it follows that we should never expect to see symbol-oriented information-processing models of any value to either area.

Indeed, this weakness of information-processing models is, according to recent attacks from the connectionists (Rumelhart and McClelland, 1986), the Achilles heel of the symbolic approach to information processing. Furthermore, from a developmental perspective, the situation is particularly troublesome, for if we are to model a system from its neonatal origins, then we will have to invent new ways to model the interface between perceptual-motor systems and central cognition, particularly at the outset, when they provide the basis for all subsequent cognition. At present, there are not enough connectionist -- or "parallel-distributed-processing" (PDP) -- models of developmental phenomena to decide the extent to which they will replace, augment, or be absorbed by the symbolic variety of information-processing models described in this chapter. Nevertheless, the connectionist criticisms of symbol-oriented approaches to cognition in general, and the more developmentally relevant points listed above, warrant careful consideration.

3.3 Self-modifying systems: Development or learning?

Recall that earlier, I side-stepped the distinction between learning and development by using the term "change" for self-modification. However, I need to return to the issue, because a common criticism of the kind of systems described above is that while

they may account for learning, they certainly do not capture the "essence" of development (cf. Beilin, 1981; Neisser, 1976). I disagree. If we look at the many dichotomies that have been used to distinguish development from learning, the self-modifying systems appear to be more appropriately placed in the development category than in the learning category.

- *Spontaneous versus imposed.* Much of development appears to occur "on its own," without any external agent instructing, inducing, or urging the change. But this is precisely the phenomenon that Siegler's strategy-choice model and Neche's HPM were designed to account for. In SOAR, chunking occurs continuously and results in changes whenever the system detects the appropriate circumstances. It has the flavor of the experience-contingent spontaneity that purportedly distinguishes development from learning.
- *Qualitative versus quantitative change.* This distinction has occupied philosophers and developmentalists for many years, and I can only suggest one modest clarification. Look at a program that has undergone self-modification, and ask whether the change is quantitative or qualitative. For example, in the Anzai and Simon (1979) work, it seems to me that the change from depth-first search to a recursive strategy could only be characterized as qualitative, and hence more of a developmental change than a learning one. Similarly, the HPM system transforms an inefficient strategy for addition (counting out the augend, counting out the addend, and then counting out the total set) into an efficient one (starting with the maximum of the two arguments and then "counting on" the other argument). It is difficult to characterize this as simply a change in which more of some pre-existing feature is added to the system: "qualitative change" seems the appropriate designation.
- *Structural reorganization versus local change.* Developmental theories, particularly those with a strong emphasis on stages (cf. Fischer, 1980), usually demand structural reorganization as a requirement for development, while viewing local changes as the province of learning. Clearly, some of the basic mechanisms in self-modifying production systems operate on a relatively local basis. Indeed, one of the great advantages of production systems is that they do not require vast systematic knowledge of the consequences of local changes. But when we begin to look carefully at changes in information-processing systems, the distinction between "local" and "structural" changes becomes blurred. Changing a few conditions in an existing production (a local change) may radically alter the firing sequence of it and all its previous successors, producing very different patterns of activation in working memory. This in turn would result in different patterns of goals and subgoals, and, ultimately, in a different set of generalizations and rules. Thus, from local changes come global effects, and from incremental modifications come structural reorganizations.
- *Reflective abstraction versus practice versus knowledge of results.* The systems described in this chapter constitute a very different class of models from earlier models of paired-associate learning (Feigenbaum, 1963) or concept learning (Gregg & Simon, 1967). Such models were clearly intended to account for learning in situations with externally supplied feedback about the

correctness of the current state of the system. In systems like HPM, or proposed systems like BAIRN (Wallace, Klahr & Bluff, 1987), change is not dependent on explicit feedback from the environment. Instead, many of the processes that seek patterns are self-contained, in the sense that they examine the trace of the system's own encodings in the absence of any clear indications of a "right" or "wrong" response. Such processes can be viewed as a mechanization of Piaget's "reflective abstraction."

- *Active or Passive?* Information-processing models have been criticized for painting "a strikingly passive picture of the child" (Liben, 1987, p. 119). While a passive model might account for learning -- especially learning from instruction -- it could not, so the argument goes, account for the active, seeking, self-initiated nature of cognitive development. But it should be clear by now that computer simulation models must, by their very nature, make explicit statements about how goals are set, how agenda's are constructed, or how self-direction is initiated or maintained.⁹ Assertions about the particular ways in which this "active" engagement with the environment occurs may well be inadequate or incorrect, but not until the creation of information-processing models was it possible to make unambiguous statements about these centrally important issues.

These dichotomies are not independent, nor do they exhaust the possible contrasts between development and learning. This listing should suffice, however, to show that at the level at which such contrasts are stated, there is little basis for the claim that information-processing models in general, or self-modifying production systems in particular, are inherently inadequate to capture the essence of cognitive development.

3.4 The future of the hard-core approach

In the early years of computer simulation, the necessary resources were limited to very few research centers. Even today, only a handful of developmental psychologists have had any extensive training with computer simulation models. However, with the widespread distribution of powerful workstations, the proliferation of computer networks for transmitting programs and systems, and the increasing number of published reports on various kinds of computationally based cognitive architectures, the appropriate technology and support structures are relatively accessible. This accessibility will make it possible to include simulation methodology as a standard part of the training of cognitive developmentalists.

The situation appears somewhat like the early days of other kinds of computational technology, such as standard statistical packages, or scaling procedures. The earliest papers using those techniques usually required many pages of description about the fundamental ideas, before the task at hand could be addressed. Today, the reader of a paper using analysis of variance or multidimensional scaling is expected to have had several courses in graduate school learning the fundamentals. Similarly, early papers on production systems all included a brief tutorial on the basic concepts, before presenting a production system model of the specific domain.

⁹Given the importance of children's "active construction of their own environment" to neo-Piagetians, it is surprising and frustrating to search in vain through Piaget's theoretical formulations for a clear statement of how any of these processes operate.

Over the next ten years, I expect to see theories of cognitive development couched in terms of extensions to systems like SOAR, or ACT*, or some other well-known (by then) cognitive architecture. The writers of those papers will be able to assume that readers need no more of a tutorial in the underlying system than current writers assume that they have to explain the conceptual foundations or computational details of an ANOVA. My vision is that, with respect to the hard-core information-processing approach to cognitive development, we will be able to expect the same level of technical training in the developmental psychologist of the future. One we are fully armed with such powerful tools, progress on our most difficult problems will be inevitable. We will no longer talk of "approaches" to our problems, but rather, of their solutions.

**Table 1: FEATURES OF INFORMATION-PROCESSING APPROACHES
TO COGNITIVE DEVELOPMENT**

Features of Soft-core information processing approaches:

THEORETICAL FEATURES

- *S1: The assumption that the child's mental activity can be described in terms of processes that manipulate symbols and symbol structures.*
- *S2: The assumption that these symbolic processes operate within an information processing system with identifiable properties, constraints, and consequences.*
- *S3: The characterization of cognitive development as self-modification of the information processing system.*

METHODOLOGICAL FEATURES

- *S4: Use of formal notational schemes for expressing complex, dynamic systems.*
- *S5: Modelling the time-course of cognitive processing over relatively short durations: chronometric analysis.*
- *S6: Use of high-density data from error-patterns and protocols to induce and test complex models.*
- *S7: Use of highly detailed analyses of the environment facing the child on specific tasks.*

Features of hard-core information processing approaches:

- *H1: Use of computer simulation.*
- *H2: Commitment to elements of the simulation as theoretical assertions, rather than just metaphor or computational convenience.*
- *H3: Goal of creating a complete self-modifying simulation that accounts for both task performance and development.*

References

- Anderson, J.R. (1976). *Language, memory, and thought*. Hillsdale, NJ: Erlbaum.
- Anderson, J.R. (Ed.). (1981). *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum.
- Anderson, J.R. (1983). *The architecture of cognition*. Cambridge MA: Harvard University Press.
- Anderson, J.R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review*, 94, 192-210.
- Anderson, J.R. & Kline, P.J. (1979). A learning system and its psychological implications. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*. Tokyo: IJCAI.
- Anderson, J.R., Greeno, J.G., Kline, P.J., & Neves, D.M. (1981). Acquisition of problem solving skill. In J.R. Anderson (Ed.), *Cognitive Skills and their acquisition*. Hillsdale, NJ: .
- Anderson, J.R., Kline, P.J., & Beasley, C.M., Jr. (1978). *A general learning theory and its application to schema abstraction* (Tech. Rep. 78-2). Department of Psychology, Carnegie-Mellon University.
- Anzai, Y., & Simon, H.A. (1979). The theory of learning by doing. *Psychological Review*, 86, 124-140.
- Ashcraft, M.H. (1982). The development of mental arithmetic: A chronometric approach. *Developmental Review*, 2(3), 213-236.
- Ashcraft, M.H. (1987). Children's knowledge of simple arithmetic: A developmental model and simulation. In J. Bisanz, C.J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research*. New York: Springer-Verlag.
- Atkinson, R.C., & Shiffrin, R.M. (1968). Human memory: A proposed system and its control processes. In K.W. Spence & J.T. Spence (Eds.), *The psychology of learning and motivation*. New York: Academic Press.
- Bakker, P.E., & Halford, G.S. (January 1988). *A basic computational theory of structure-mapping in analogy and transitive inference* (Tech. Rep.). Centre for Human Information Processing and Problem Solving, University of Queensland, Australia.
- Banks, M.S. (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.
- Banks, M.S. (1988). Visual recalibration and the development of contrast and optical flow perception. In A. Yonas (Ed.), *Perceptual Development in Infancy*. Hillsdale, NJ: Erlbaum.
- 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.

- Baylor, G.W., Gascon, J., Lemoyne, G., & Pother, N. (1973). An information processing model of some seriation tasks. *Canadian Psychologist*, 14, 167-196.
- Beilin, H. (May 1981). Piaget and the new functionalism. In . Address given at the Eleventh Symposium of the Piaget Society, Philadelphia.
- Beilin, H. (1983). The new functionalism and Piaget's program. In E.K. Scholnick (Ed.), *New trends in conceptual representation. Challenges to Piaget's theory?*. Hillsdale, NJ: Erlbaum.
- Beilin, H. (1987). Selective perspectives on cognitive/intellectual development. *Contemporary Psychology*, 32(12), 1010-1012. Review of M. Perlmutter, 'Perspectives on intellectual development: The Minnesota symposia on child psychology', Vol. 19.
- Berliner, H., & Ebeling, C. (1988). Pattern knowledge and search: The SUPREM architecture. *Artificial Intelligence*, . In press.
- Bisanz, J., Brainerd, C.J., & Kail, R. [Eds.]. (1987). *Formal methods in developmental psychology: Progress in cognitive development research*. New York: Springer-Verlag.
- Brainerd, C.J. (1987). Structural measurement theory and cognitive development. In J. Bisanz, C.J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research*. New York: Springer-Verlag.
- Brown, A.L. (1982). Learning and development: The problem of compatibility, access and induction. *Human Development*, 25, 89-115.
- Brown, J.S., & Burton, R.R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2, 155-192.
- Brown, J.S., & VanLehn, K. (1982). Towards a generative theory of 'bugs'. In T. Romberg, T. Carpenter, & J. Moser (Eds.), *Addition and subtraction: A developmental perspective*. Hillsdale, NJ: Erlbaum.
- Brownston, L., Farrell, R., Kant, E., & Martin, N. (1985). *Programming expert systems in OPS5: An introduction to rule-based programming*. Reading, MA: Addison-Wesley.
- Card, S.K., Moran, T., & Newell, A. (1983). *Applied information processing psychology: The human-computer interface*. Hillsdale, N.J.: Erlbaum.
- Case, R. (1985). *Intellectual development: Birth to adulthood*. 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*. Hillsdale, NJ: Erlbaum.
- Celle'rier, G. (1972). Information processing tendencies in recent experiments in cognitive learning-theoretical implications. In S. Farnham-Diggory (Ed.), *Information processing in children*. New York: Academic Press.
- Chase, W.G. (Ed.). (1973). *Visual information processing*. New York: Academic Press.

- Chase, W.G. (1978). Elementary information processes. In W.K. Estes (Ed.), *Handbook of learning and cognitive processes*, Vol. 5. Hillsdale, NJ: Erlbaum.
- Chi, M.T.H. (1976). Short-term memory limitations in children: Capacity or processing deficits? *Memory and Cognition*, 4, 559-572.
- Chi, M.T.H. (1977). Age differences in memory span. *Journal of Experimental Child Psychology*, 23, 266-281.
- Chi, M.T.H. (1978). Knowledge structures and memory development. In R. Siegler (Ed.), *Children's thinking: What develops?*. Hillsdale, NJ: Erlbaum.
- Chi, M.T.H. (1981). Knowledge development and memory performance. In M. Friedman, J.P. Das, & N. O'Connor (Eds.), *Intelligence and learning*. New York: Plenum Press.
- Chi, M.T.H. & Klahr, D. (1975). Span and rate of apprehension in children and adults. *Journal of Experimental Child Psychology*, 19, 434-439.
- 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.
- Davidson, J.E. (1986). The role of insight in giftedness. In R.J. Sternberg & J.E. Davidson (Eds.), *Conceptions of giftedness*. New York: Cambridge University Press.
- DeLoache, J.S. (1987). Rapid change in the symbolic functioning of very young children. *Science*, 238, 1556-1557.
- DeLoache, J.S. (1988). The development of representation in young children. In H.W. Reese (Ed.), *Advances in Child Development and Behavior*. New York: Academic Press. In press.
- Dunbar, K., & Klahr, D. (1988). Developmental differences in scientific discovery strategies. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon*. Hillsdale, NJ: Erlbaum. In press.
- Ericsson, A. & Simon, H.A. (1984). *Protocol Analysis: Verbal Reports as Data*. Cambridge, MA: MIT Press.
- Falkenhainer, B., Forbus, K.D., & Gentner, D. (1986). The structure-mapping engine. In *Proceedings of the American Association for Artificial Intelligence*. Philadelphia: American Association for Artificial Intelligence.
- Farnham-Diggory, S. [Ed.]. (1972). *Information processing in children*. New York: Academic Press.
- Fay, A.L., & Mayer, R.E. (1987). Children's naive conceptions and confusions about LOGO graphics commands. *Journal of Educational Psychology*, 79(3), 254-268.
- Feigenbaum, E. A. (1963). The simulation of verbal learning behavior. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*. New York: McGraw-Hill.
- Feigenbaum, E.A., & Simon, H.A. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, 8, 305-336.

- Fischer, K.W. (1980). A theory of cognitive development: The control and construction of hierarchies of skills. *Psychological Review*, 87, 477-531.
- Fischer, K.W., & Pipp, S.L. (1984). Processes of cognitive development: Optimal level and skill acquisition. In R.J. Sternberg (Ed.), *Mechanisms of cognitive development*. New York: Freeman.
- Flavell, J.H. (1982). Structures, stages, and sequences in cognitive development. In W.A. Collins (Ed.), *The concept of development*. Hillsdale, NJ: Erlbaum.
- 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*. New York: Oxford University Press.
- Geary, D.C., Widaman, K.F., Little, T.D., & Cormier, P. (July 1987). Cognitive addition: Comparison of learning disabled and academically normal elementary school children. *Cognitive Development*, 2(3), 249-270.
- Gentner, D. (February 1988). Metaphor as structure mapping: The relational shift. *Child Development*, 59(1), 47-59.
- Goodman, N. (1968). *Languages of art*. Indianapolis: Bobbs-Merrill.
- Greeno, J.G., Riley, M.S., & Gelman, R. (1984). Conceptual competence and children's counting. *Cognitive Psychology*, 16(1), 94-143.
- Gregg, L. W., & Simon, H. A. (1967). Process models and stochastic theories of simple concept formation. *Journal of Mathematical Psychology*, 4, 246-276.
- Groen, G.J., & Parkman, J.M. (1972). A chronometric analysis of simple addition. *Psychological Review*, 79, 329-343.
- Hagen, J.W. (1972). Strategies for remembering. In S. Farnham-Diggory (Ed.), *Information processing in children*. New York: Academic Press.
- Halford, G. S. (1970). A theory of the acquisition of conservation. *Psychological Review*, 77, 302-316.
- Halford, G. S. (1975). Children's ability to interpret transformations of a quantity. I: An operational system for judging combinations of transformations. *Canadian Journal of Psychology*, 29, 124-141.
- Hill, J.C. (1983). A computational model of language acquisition in the two-year-old. *Cognition and Brain Theory*, 6, 287-317.
- 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*. Los Altos, CA: Kaufmann.
- Hoyer, W.J., & Familant, M.E. (1987). Adult age differences in the rate of processing expectancy information. *Cognitive Development*, 2(1), 59-70.
- Hunter, I.M.L. (1968). Mental calculation. In P.C. Wason & P.N. Johnson-Laird (Eds.), *Thinking and reasoning*. Baltimore, MD: Penguin Books.

- Inhelder, B. (1972). Information processing tendencies in recent experiments in cognitive learning-empirical studies. In S. Farnham-Diggory (Ed.), *Information processing in children*. New York: Academic Press.
- Just, M.A., & Carpenter, P.A. (1978). Inference processes during reading: Reflections from eye fixations. In J.W. Senders, D.F. Fisher & R.A. Monty (Eds.), *Eye movements and the higher psychological functions*. Hillsdale, NJ: Erlbaum.
- Kail, R. (1984). *The development of memory in children: Second edition*. New York: Freeman.
- Kail, R. (1988). Developmental functions for speeds of cognitive processes. *Journal of Experimental Child Psychology*, 45, 339-364.
- Kail, R. & Bisanz, J. (1982). Information processing and cognitive development. In H.W. Reese (Ed.), *Advances in child development and behavior*. New York: Academic Press.
- Kail, R., Pellegrino, J., & Carter, P. (1980). Developmental changes in mental rotation. *Journal of Experimental Child Psychology*, 29, 102-116.
- Keating, D.P., & Bobbitt, B.L. (1978). Individual and developmental differences in cognitive processing components of mental ability. *Child Development*, 49, 155-167.
- Kellman, P.H. (1988). Theories of perception and research in perceptual development. In A. Yonas (Ed.), *Perceptual development in infancy*. Hillsdale, NJ: Erlbaum.
- Kintsch, W., & Greeno, J.G. (1985). Understanding and solving word arithmetic problems. *Psychological Review*, 92, 109-129.
- Klahr, D. (1973). A production system for counting, subitizing, and adding. In W.G. Chase (Ed.), *Visual information processing*. New York: Academic Press.
- Klahr, D., & Carver, S.M. (1988). Cognitive objectives in a LOGO debugging curriculum: Instruction, Learning, and Transfer. *Cognitive Psychology*, . In Press.
- Klahr, D., & Robinson, M. (1981). Formal assessment of problem solving and planning processes in preschool children. *Cognitive Psychology*, 13, 113-148.
- 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. New York: Academic Press.
- Klahr, D., & Wallace, J. G. (1970). The development of serial completion strategies: An information processing analysis. *British Journal of Psychology*, 61, 243-257.
- Klahr, D., & Wallace, J. G. (1970). An information processing analysis of some Piagetian experimental tasks. *Cognitive Psychology*, 1, 358-387.
- Klahr, D., & Wallace, J. G. (1972). Class inclusion processes. In S. Farnham-Diggory (Ed.), *Information processing in children*. New York: Academic Press.
- Klahr, D., & Wallace, J. G. (1973). The role of quantification operators in the development of conservation of quantity. *Cognitive Psychology*, 4, 301-327.
- Klahr, D., & Wallace, J.G. (1976). *Cognitive development: An information processing view*. Hillsdale, NJ: Erlbaum.

- Klahr, D., Langley, P., & Neches, R. [Eds.]. (1987). *Production systems models of learning and development*. Cambridge, MA: MIT Press.
- 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.
- Kuhn, D., & Phelps, E. (1982). The development of problem solving strategies. In H.W. Reese (Ed.), *Advances in child development and behavior*. New York: Academic Press.
- Lachman, R., Lachman, J., & Butterfield, E.C. (1979). *Cognitive psychology and information processing: An introduction*. Hillsdale, NJ: Erlbaum.
- Langley, P. (1982). Language acquisition through error recovery. *Cognition and Brain Theory*, 5, 211-255.
- Langley, P. (1987). A general theory of discrimination learning. In D. Klahr, P. Langley, & R. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.
- Langley, P., Neches, R., Neves, D.M., & Anzai, Y. (1980). A domain-independent framework for procedure learning. *Policy Analysis and Information Systems*, 4(2), 163-197.
- 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*. Hillsdale, NJ: Erlbaum.
- Lewis, C.H. (1978). *Production system models of practice effects*. Doctoral dissertation, University of Michigan.
- Lewis, C. (1981). Skill in algebra. In J. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum.
- Lewis, C. (1987). Composition of productions. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.
- Liben, L.S. [Ed.]. (1987). *Development and learning: Conflict or congruence?*. Hillsdale, NJ: Erlbaum.
- Mackworth, A.K. (1987). What is the schema for a schema? *Behavioral and Brain Sciences*, 10(3), 443-444.
- MacWhinney, B.J. [Ed.]. (1987). *Mechanisms of language acquisition*. Hillsdale, NJ: Erlbaum.
- MacWhinney, B. (1988). *Competition and cooperation in language processing*. In press.
- Madden, D.J. (1987). Aging, attention, and the use of meaning during visual search. *Cognitive Development*, 2(3), 201-216.
- Mandler, J.M. (1983). Representation. In P.H. Mussen (Ed.), *Cognitive Development*. Vol. III: *Handbook of Child Psychology*. New York: Wiley.

- Mandler, J.M., & Johnson, N.S. (1977). Rememberance of things parsed: Story structure and recall. *Cognitive Psychology*, 9, 111-151.
- Marr, D. (1982). *Vision A computational investigation into the human representation and processing of visual information*. San Francisco: Freeman.
- Miller, P.H. (1983). *Theories of developmental psychology*. San Francisco: Freeman.
- Naus, M.J., & Ornstein, P.A. (1983). Development of memory strategies: Analysis, questions, and issues. In M.T. Chi (Ed.), *Trends in memory development research*. New York: Karger.
- Neches, R. (1981). *Models of heuristic procedure modification*. Doctoral dissertation, Department of Psychology, Carnegie-Mellon University.
- Neches, R. (1982). Simulation systems for cognitive psychology. *Behavior Research Methods and Instrumentation*, 14, 77-91.
- Neches, R. (1987). Learning through incremental refinement of procedures. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.
- 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.
- Neisser, U. (1976). General, academic and artificial intelligence. In L.B. Resnick (Ed.), *The nature of intelligence*. Hillsdale, NJ: Erlbaum.
- Nelson, K., & Gruendel, J.M. (1981). Generalized event representations: Basic building blocks of cognitive development. In M.E. Lamb & A.L. Brown (Eds.), *Advances in developmental psychology*, Vol. 1. Hillsdale, NJ: Erlbaum.
- Neves, D. & Anderson, J.R. (1981). Knowledge compilation: Mechanisms for the automatization of cognitive skills. In J.R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum.
- Newell, A. (1966). On the representation of problems. *Computer science research review*, 18-33.
- Newell, A. (1968). On the analysis of human problem solving protocols. In J. C. Gardin & B. Javlin (Eds.), *Calcul et formalization dan les sciences de l'homme*. Center de la Recherche Scientifique.
- Newell, A. (1972). A note on process-structure distinctions in developmental psychology. In S. Farnham-Diggory (Ed.), *Information processing in children*. New York: Academic Press.
- Newell, A. (1973). Production systems: Models of control structures. In W.G. Chase (Ed.), *Visual information processing*. 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. Hillsdale, NJ: Erlbaum.

- Newell, A. (1988). Putting it all together: Final comments. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon*. Hillsdale, NJ: Erlbaum. [b].
- Newell, A. (1988). The 1987 William James Lectures: Unified theories of cognition. Departments of Computer Science and Psychology, Carnegie-Mellon University.
- Newell, A., & McDermott, J. (1975). *PSG Manual*. Pittsburgh: Carnegie-Mellon University, Department of Computer Science.
- Newell, A., & Simon, H. A. (1972) *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.
- Nisbett, R.E., & Wilson, T.D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, 84, 231-259.
- Norman, D.A., Rumelhart, D.E., & the LNR Research Group. (1975). *Explorations in cognition*. San Francisco, CA: Freeman.
- 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*. Hillsdale, NJ: Erlbaum.
- Pollack, R.H. (1972). Perceptual development: A progress report. In S. Farnham-Diggory (Ed.), *Information Processing in Children*. New York: Academic Press.
- Rabinowitz, F.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*. New York: Springer-Verlag.
- Resnick, L.B., & Glaser, R. (1976). Problem solving and intelligence. In L.B. Resnick (Ed.), *The nature of intelligence*. Hillsdale, NJ: Erlbaum.
- Riley, M.S., Greeno, J.G., & Heller, J.I. (1983). Development of children's problem-solving ability in arithmetic. In H.P. Ginsburg (Ed.), *The development of mathematical thinking*. New York: Academic Press.
- Rosenbloom, P.S., & Newell, A. (1982). Learning by chunking: Summary of a task and a model. In *Proceedings of the Second National Conference on Artificial Intelligence*. Los Altos, CA: Morgan-Kaufmann.
- 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*. Cambridge, MA: MIT Press.
- Rumelhart, D.E., & McClelland, J.L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA: The MIT Press.
- Rumelhart, D.E. & Norman, D.A. (1981). Analogical processes in learning. In J.R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum.
- Shultz, T.R. (1987). A computational model of judgments of causation, responsibility, blame, and punishment. Based on a paper delivered to the Meetings of the Society for Research in Child Development, Baltimore, April 1987.

- Shultz, T.R. (1988). Making causal judgments. Department of Psychology, McGill University.
- Siegler, R.S. (1976). Three aspects of cognitive development. *Cognitive Psychology*, 8(4), 481-520.
- Siegler, R.S. (1981). Developmental sequences within and between concepts. *Monographs of the Society for Research in Child Development*, vol. 46, Whole No. 189.
- Siegler, R.S. (1983). Information processing approaches to development. In P.H. Mussen (Ed.), *Volume 1. History, Theory, and Methods. Handbook of Child Psychology*. New York: John Wiley & Sons.
- Siegler, R.S. (1986). Unities in strategy choices across domains. In M. Perlmutter (Ed.), *Minnesota symposium on child psychology*. Hillsdale, NJ: Erlbaum.
- Siegler, R.S. (1988). The perils of averaging data over strategies: An example from children's addition. *Journal of Experimental Psychology: General*. In press.
- Siegler, R.S. (1989). Mechanisms of cognitive development. *Annual Review of Psychology*. pp. In press.
- Siegler, R.S., & Shipley, C. (1987). The role of learning in children's strategy choices. In L. S. Liben (Ed.), *Development and learning: Conflict or congruence?*. Hillsdale, NJ: Erlbaum.
- Simon, H. A. (1962). An information processing theory of intellectual development. *Monographs of the Society for Research in Child Development*, 27(2), Serial No. 82.
- Simon, H. A. (1972). On the development of the processor. In S. Farnham-Diggory (Ed.), *Information processing in children*. New York: Academic Press.
- Simon, D.P., & Simon, H.A. (1978). Individual differences in solving physics problems. In R. Siegler (Ed.), *Children's thinking. What develops?*. Hillsdale, NJ: Erlbaum.
- Sperling, G. (1960). The information available in brief visual presentations. *Psychological Monographs*, 74, (11, Whole No. 498).
- Spitz, H.H., & Borys, S.V. (1984). Depth of search: How far can the retarded search through an internally represented problem space? In P.H. Brooks, R. Sperber, & C. McCauley (Eds.), *Learning and cognition in the mentally retarded*. Hillsdale, NJ: Erlbaum.
- Stein, N.L., & Glenn, C.G. (1979). An analysis of story comprehension in elementary school children. In R.O. Freedle (Ed.), *New directions in discourse processing*, Vol. 2. Norwood, NJ: Ablex.
- Sternberg, S. (1966). High speed scanning in human memory. *Science*, 153, 652-654.
- Sternberg, R.J. [Ed.]. (1984). *Mechanisms of cognitive development*. New York: Freeman.
- Sternberg, R.J., & Rifkin, B. (1979). The development of analogical reasoning processes. *Journal of Experimental Child Psychology*, 27, 195-232.

- Strauss, S. [Ed.]. (1982). *U-shaped behavioral growth*. New York: Academic Press.
- Wallace, J.G., Klahr, D. & Bluff, K. (1987). A self-modifying production system of cognitive development. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.
- Waterman, D. (1975). Adaptive production systems. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*. Cambridge, MA: Artificial Intelligence Laboratory, MIT.
- Wilkinson, A.C. & Haines, B.A. (1987). Learning a cognitive skill and its components. In J. Bisanz, C.J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research*. New York: Springer-Verlag.
- Yonas, A. [Ed.]. (1988). *Perceptual development in infancy*. Hillsdale, NJ: Erlbaum.
- Young, R.M. (1976). *Seriation by children. An artificial intelligence analysis of a Piagetian task*. Basel: Birkhauser.
- Young, R.M., & O'Shea, T. (1981). Errors in children's subtraction. *Cognitive Science*, 5, 153-177.