COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 35 (2011) 1329–1351

ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/j.1551-6709.2011.01190.x

Criteria for the Design and Evaluation of Cognitive Architectures

Sashank Varma

Department of Educational Psychology, University of Minnesota

Received 30 April 2009; received in revised form 18 February 2011; accepted 21 February 2011

Abstract

Cognitive architectures are unified theories of cognition that take the form of computational formalisms. They support computational models that collectively account for large numbers of empirical regularities using small numbers of computational mechanisms. *Empirical coverage* and *parsimony* are the most prominent criteria by which architectures are designed and evaluated, but they are not the only ones. This paper considers three additional criteria that have been comparatively undertheorized. (a) Successful architectures possess *subjective* and *intersubjective meaning*, making cognition comprehensible to individual cognitive scientists and organizing groups of like-minded cognitive scientists into genuine communities. (b) Successful architectures provide *idioms* that structure the design and interpretation of computational models. (c) Successful architectures are *strange*: They make provocative, often disturbing, and ultimately compelling claims about human information processing that demand evaluation.

Keywords: Artificial intelligence; Psychology; Cognitive architecture; Philosophy of computation; Philosophy of science; Computer simulation; Neural networks; Symbolic computational modeling

1. Introduction

This paper is about the design and evaluation of cognitive architectures. Cognitive architectures are unified theories of cognition that take the form of computational formalisms (Newell, 1990). As unified theories, they are responsible for all the phenomena of cognition. As computational formalisms, they embody the view that cognition is a form of information processing. Perhaps the best-known architecture is ACT-R (Anderson, 2007), which

Correspondence should be sent to Sashank Varma, Educational Psychology, 165 Education Sciences Building, 56 East River Rd, Minneapolis, MN 55455. E-mail: sashank@umn.edu

combines symbolic, activation-based, and Bayesian computational mechanisms, but there are others.

The most prominent criteria by which cognitive architectures are designed and evaluated are *empirical coverage* and *parsimony*. Successful architectures provide broad empirical coverage of cognition: They account for large numbers of empirical regularities spread across many domains. For example, the 4CAPS architecture supports models of sentence comprehension, problem solving, visuospatial reasoning, and dual tasking (Just & Varma, 2007). Successful architectures are also parsimonious: They consist of relatively small numbers of computational mechanisms. Cognitive scientists design architectures based on these criteria, identifying minimal sets of computational mechanisms that support a maximal number of domains models. Cognitive scientists also evaluate architectures based on these criteria: Given empirically equivalent architectures, they prefer the one that is more parsimonious, and given equally parsimonious architectures, they prefer the one with the greater empirical coverage.

Although empirical coverage and parsimony are important criteria for the design and evaluation of cognitive architectures, they are not the only ones. This paper considers three additional criteria that have been comparatively undertheorized. (a) Successful architectures possess *subjective* and *intersubjective meaning*, making cognition comprehensible to individual cognitive scientists and organizing groups of like-minded cognitive scientists into genuine communities. (b) Successful architectures provide *idioms* that structure the design and interpretation of computational models. (c) Successful architectures are *strange*: They make provocative, often disturbing, and ultimately compelling claims about human information processing that demand evaluation. These criteria often operate implicitly in the design and evaluation of architectures. This paper makes them explicit, examining their role in the history and practice of cognitive science.

This paper first describes cognitive architectures and the conventional criteria by which they are designed and evaluated. It then develops the additional criteria of subjective and intersubjective meaning, idiom-driven design, and strangeness. It concludes by calling for further investigations of the criteria that shape theorizing in cognitive science.

2. Conventional cognitive architectures

2.1. Domain models and their discontents

The bulk of cognitive science research is carried out in domain-specific trenches. Cognitive scientists focus on domains such as visual perception, sentence comprehension, and problem solving. They formulate theories and test them against empirical regularities discovered in experiments. A hallmark of cognitive science is that many theories take the form of computational models. Models are organizations of computational mechanisms: representations, operations on representations, and control structures for specifying the application of operations to representations over time. For example, a production system model of sentence comprehension might encode phrase markers as declarative memory elements and

phrase-structure rules as productions, and use parallel processing to parse sentences—to apply phrase-structure rules to phrase markers to generate new phrase markers—in a bottom-up manner (e.g., Just & Carpenter, 1992).

Computational models are designed and evaluated against the criteria of empirical coverage and parsimony. Consider a model of a domain. Its empirical coverage is the correspondence between its temporal profiles, error patterns, and activation fluctuations and those of humans. Empirical coverage is of course the first criterion of scientific theories: In a good theory, ''there is an element corresponding to each element of reality'' (Einstein, Podolsky, & Rosen, 1935, p. 777). A model's parsimony is the efficiency with which it covers its domain. Successful models cover their domains in an efficient manner, accounting for relatively many empirical regularities using relatively few computational mechanisms. Fig. 1A shows a slice through the space of models.

There are two problems with theorizing entirely at the level of domain-specific computational models. The *problem of unification* threatens empirical coverage (Newell, 1990, pp. 17–18). It is the goal of every science to give a unified account of all its phenomena (Neurath, Carnap, & Morris, 1955; Oppenheim & Putnam, 1958). Cognitive science, like all sciences, strives for unified theories—for comprehensive accounts that range from sensation up to rational thought. If cognitive scientists work in domain-specific silos, with some developing models of visual attention, others models of categorization, and still others models of language comprehension, then we are unlikely to get unified theories of all cognition. In fact, over time, models of different domains will come to resemble each other less and less. This is a consequence of the falsificationalist strategy of psychological experimentation

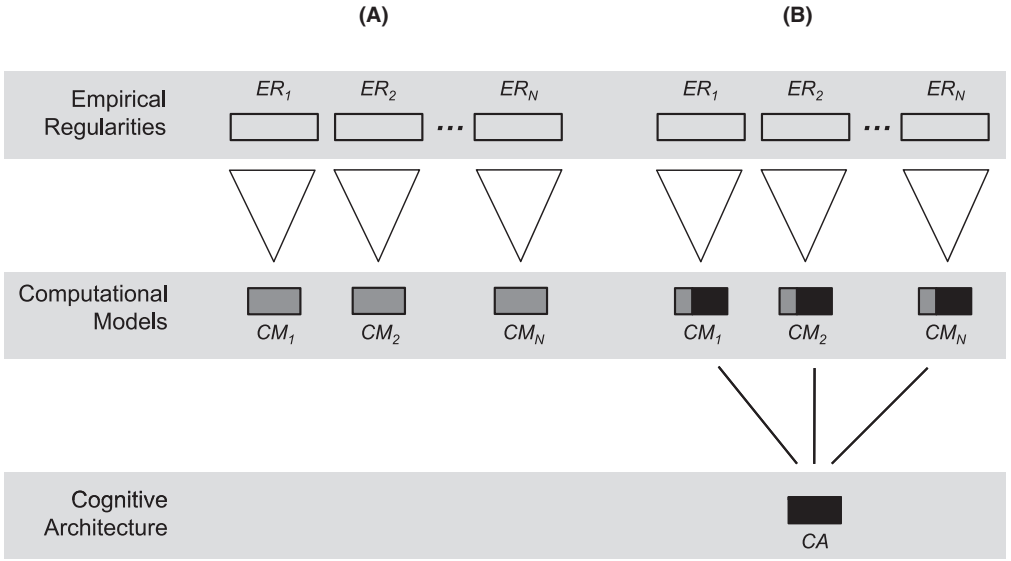

Fig. 1. (A) Computational models (CM) account for empirical regularities (ER) within domains (indexed by *i*). (B) A cognitive architecture (CA) factors out domain-general computational mechanisms, enabling more parsimonious coverage of empirical regularities across domains.

(Popper, 1963), which focuses on finding differentiating phenomena within domains, not unifying patterns across domains (Newell, 1973a).

The *problem of degrees of freedom* threatens parsimony in domain-specific models. The empirical regularities of a domain radically underdetermine the set of possible computational models (Newell, 1990, pp. 21–22; Pylyshyn, 1984, pp. 105–106). Given the empirical regularities of syntactic parsing, for example, it is unclear whether phrase-structure rules are applied in a serial or parallel fashion. Different models are free to assume different control structures for sentence comprehension, and there is little empirical basis for choosing between them (Townsend, 1974). There is a second way in which the problem of degrees of freedom leads to models that lack parsimony. When faced with recalcitrant empirical regularities, it is tempting to tack on ad hoc computational mechanisms that account for them, but have little chance of generalizing to other empirical regularities or domains (Anderson, 1990; Newell, 1973a). Over time, models become encrusted with ad hoc mechanisms, decreasing their parsimony.

2.2. Cognitive architectures

Cognitive architectures help solve the problem of unification and the problem of degrees of freedom. An architecture is a privileged collection of computational mechanisms that forms a substrate for all of human information processing (Anderson, 1990; Marr, 1982; Newell, 1990; Pylyshyn, 1984; Rumelhart & McClelland, 1986). The mechanisms are domain-general, intended to apply across a range of phenomena. They combine with domain-specific mechanisms to form models of particular domains.[1] The relation between architectures, models, and empirical regularities is shown in Fig. 1B.

Cognitive architectures cluster into paradigms based on shared computational mechanisms. The production system paradigm includes 4CAPS, ACT-R, Soar (Newell, 1990), and EPIC (Meyer & Kieras, 1997). These architectures have generally been applied to high-level domains of cognition, and 4CAPS and ACT-R are increasingly being applied to phenomena in cognitive neuroscience. The connectionist paradigm includes recurrent networks, which have been applied to cognitive and language development (Elman et al., 1996), and the Leabra architecture, which has been applied to phenomena in cognitive neuroscience (O'Reilly & Munakata, 2000). The exemplar paradigm has been applied to low-level cognitive domains such as automaticity, episodic memory, categorization, decision making, and phonological processing (Dougherty, Gettys, & Ogden, 1999; Hintzman, 1986; Logan, 1988; Medin & Schaffer, 1978; Raaijmakers & Shiffrin, 1981; Stanfill & Waltz, 1986).

Theorizing at the level of cognitive architecture mitigates the problems that plague computational models. Architectures address the problem of unification by serving as common foundations on which to construct models of different domains. For example, there exist ACT-R models of phenomena ranging from the Stroop effect (Lovett, 2005) to sentence comprehension (Lewis & Vasishth, 2005) to geometry theorem proving (Kao et al., unpublished data). Because these models draw on a common set of domain-general mechanisms, they can in principle be unified with one another and with ACT-R models of other domains to provide empirical coverage of large tracts of cognition.

Cognitive architectures address the problem of degrees of freedom by constraining the construction of computational models. When designing a model in isolation, one is free to ''pull out of an infinite grab bag of mechanisms'' those ''bizarre creations whose only justification is that they predict the phenomena'' (Anderson, 1990, p. 7). By contrast, when designing a model within an architecture, one is limited to the domain-general computational mechanisms the architecture provides, and to domain-specific mechanisms that are consistent or ''compliant'' with them (Howes & Young, 1997). This reduces the available degrees of freedom (Pylyshyn, 1984, pp. 105–106). For example, ACT-R models *must* inherit the serial control structure posited by the ACT-R architecture; they are not free to assume a parallel control structure. This constraint has produced sharper predictions about when people can and cannot achieve perfect time sharing (Byrne & Anderson, 2001). The more a model relies on domain-general (vs. domain-specific) mechanisms, the greater its parsimony. This is because the cost of domain-general mechanisms is amortized across models (Newell, 1990, p. 22).

3. Subjective and intersubjective meaning

Successful cognitive architectures possess subjective and intersubjective meaning: They make cognition comprehensible to individual cognitive scientists, and they organize like-minded cognitive scientists into genuine communities. These meaning dimensions are not emphasized in conventional analyses of architectures (Anderson, 1990; Marr, 1982; Newell, 1990; Pylyshyn, 1984; Rumelhart & McClelland, 1986). However, they are consistent with philosophies of science that portray graduate training as enculturating budding scientists into scientific communities (Kuhn, 1996). Through this training, scientists acquire the paradigmatic principles of a theoretical *worldview* (Goodman, 1968; Hanson, 1958) that provides a ''group-licensed way of seeing'' (Kuhn, 1996, p. 189). Architectural worldviews structure the computational, theoretical, and empirical research activities of cognitive scientists.

3.1. Architectural worldviews

Understanding the subjective meaning of a cognitive architecture requires going beyond reading scientific papers. It requires internalizing the architecture's worldview—the distinctive information processing style it attributes to cognition. Tutorials must be worked through, modeling exercises completed, and simulation environments mastered. This can require a year or more of sustained effort, typically spent during graduate school or during a post-doctoral or sabbatical year. Through these efforts, cognitive scientists learn to see cognition through the architecture's worldview (Wittgenstein, 1958, pp. 194–197) and to design computational models that embody its characteristic information processing style.

Understanding the intersubjective meaning of a cognitive architecture means understanding the research activities of other cognitive scientists who share the same architectural worldview. By apprenticing in an architectural community, cognitive scientists learn from conversations with other members of the community. They inspect and adapt computational

models written by others, and they share their own models and modify them based on feedback provided by more experienced members of the community (Newell, 1990, pp. 504–505). In this way, an architecture becomes ''a language spoken among all members of the community, rather than a language spoken by authors of the theory to readers of the theory'' (Anderson, 2007, p. 41).

The development of subjective and intersubjective meaning is illustrated by the growth of the Soar architecture and community. Through the mid-1980s, Soar was a project local to Carnegie Mellon University, to the tight research group that formed around Newell, Laird, and Rosenbloom. Soar 4 was released in 1986 with the explicit goal of attracting a larger architectural community (Laird & Rosenbloom, 1996, p. 31). The architecture was ported to Common Lisp so that it could run on a range of workstations, the user interface was improved, and a manual was written. These changes made it possible for cognitive scientists at other institutions to use Soar, and to develop a subjective understanding of it. To foster intersubjective understanding of Soar, regular workshops were organized. (As of 2010, 30 have been held.) Soar developers and users also formed a virtual community, knitted together by email and other communication technologies (Carley & Wendt, 1988). The growth and maintenance of the Soar community was important enough that Newell was willing to pay large ''managerial costs,'' spending ''over half of every day on it'' (Newell quoted in Agre, 1993, p. 442).

An expansion of subjective and intersubjective meaning also accompanied the re-emergence of connectionist architectures in the 1980s. The availability of the Parallel Distributed Processing tutorial and simulation software (McClelland & Rumelhart, 1988) enabled cognitive scientists working alone at their personal computers to move beyond reading about the models of others and to construct their own models, developing their subjective understanding of connectionist architectures. The founders of the connectionist revolution were cognizant of the need to grow to an architectural community. They organized conferences such as *Neural Information Processing Systems* (first held in 1987) and workshops such as the *Connectionist Models Summer School* (first held in 1986). This enabled cognitive scientists outside the University of California—San Diego and other hotspots ''to make contact with their colleagues,'' and to develop an intersubjective understanding of connectionist architectures (Mozer, Smolensky, Touretzky, Elman, & Weigand, 1994).

Finally, the success of ACT-R has been characterized by the growth of subjective and intersubjective meaning (Anderson, 2007, p. 41). The monograph that introduced ACT-R (Anderson, 1993) included a Common Lisp implementation of the architecture, a reference manual, and a tutorial for beginners. The software has been ported to a number of platforms over the years and the documentation refined. The widespread availability of these materials enables cognitive scientists working alone at their computers to develop a subjective understanding of ACT-R. A number of institutional structures have also been established, and around them a genuine ACT-R community has grown. An annual summer school trains new members in proper usage of the architecture, and an annual workshop speeds dissemination of domain models throughout the community. The community gains additional coherence through internet-based technologies: A website serves as a repository for its documentation, papers, and models; a mailing list enables newcomers to query experienced users about

problems that arise during model construction; and so on. These pedagogical materials and institutional structures make it possible for cognitive scientists to develop a deeper subjective and intersubjective understanding of ACT-R than was possible of its predecessors.

3.2. Generativity

The subjective and intersubjective meaning of cognitive architectures contribute to the worldviews they offer on human information processing. These worldviews are selective, magnifying some aspects of cognition for closer inspection while shrinking others into the infinite distance. With this selectivity comes *generativity*: fresh perspectives on cognitive phenomena, and their explanation as computational phenomena. Generative architectures enable cognitive scientists to glimpse ''the subtlest and most esoteric of the phenomena'' (Kuhn, 1996, p. 164). The generativity of architectures has been discussed by other cognitive scientists (Newell, 1990, p. 14; Pylyshyn, 1984, p. 128). For example, Anderson (1983) downplays conventional evaluative criteria such as the empirical coverage an architecture offers and instead emphasizes ''the success, or fruitfulness, of the theories it generates'' (p. 12).

Generativity takes two forms. *Prospectively* generative architectures open up new ground for exploration. For example, consider the rise of symbolic architectures during the 1960s and 1970s. These architectures offered fresh perspectives on high-level forms of cognition, such as the notion of problem solving as heuristic search through problem spaces (Newell & Simon, 1972). The symbolic worldview also supplied empirical methods for documenting new phenomena. One example is protocol analysis, which enables researchers to track how participants explore problem spaces as they solve problems, comprehend discourse, and so on (Newell & Simon, 1972; Pressley & Afflerbach, 1995). The result was a spate of new models, theories, and empirical regularities (Ericsson & Simon, 1993).

Retrospectively generative architectures offer new perspectives on familiar terrain. Looking at well-known empirical regularities through a new architectural worldview often suggests novel theoretical and computational accounts. Consider the explosion of connectionist architectures during the 1980s. A number of unexplained empirical regularities had accumulated during the first part of this decade, for example, from the emerging literature on the cognitive impairments of neuropsychological patients (Caplan, Baker, & Dehaut, 1985; Cohen & Squire, 1980; Warrington & Shallice, 1984). The symbolic architectures that dominated cognitive science at the time failed to offer insightful accounts of these findings. In this vacuum, the generativity of the connectionist worldview was breathtaking. For a time, it seemed as if skilled connectionists needed only glance at a domain, such as the cognitive impairments of people with dyslexia (Plaut & Shallice, 1993) or schizophrenia (Cohen & Servan-Schreiber, 1992), to generate fresh computational insights.

4. Idiom-Driven Design

Successful cognitive architectures support the design of computational models. Cognitive scientists do not generally design at the level of the computational mechanisms an

architecture provides—that would be too cumbersome (Simon, 1996). Rather, they work at a higher level, weaving together *patterns* of computation mechanisms, where each pattern serves a functional role (Gamma, Helm, Johnson, & Vlissides, 1995). Conventional analyses posit two classes of patterns: data structures formed by patterns of basic representations, and algorithms formed by patterns of basic operations. For example, Marr (1982, pp. 24–25) places at the middle level ''the representation for the input and output and the algorithm to be used to transform one into the other.'' Newell (1989, p. 404) describes the symbol level using similar terms: ''data structures with symbolic operations on them, being carried out under the guidance of plans, programs, procedures, or methods.'' Understood in this way, patterns carry a sense of the timeless a priori, like mathematical theorems. We refer to patterns as idioms to emphasize their subjective and intersubjective meaning—how they help cognitive scientists design models and understand models designed by others.

4.1. Pragmatic value

When constructing computational models within a cognitive architecture, some problems occur over and over again. Each problem can typically be solved in multiple ways, using different patterns of computational mechanisms. The choice of which pattern to use is a local form of the degrees of freedom problem. An idiom is a pattern of computational mechanisms that solves a recurring problem in a manner consistent with an architecture's characteristic information processing style. Idioms are recognized by members of the architecture's community as preferred solutions. Architectures accumulate libraries of idioms. Models constructed by combining accepted idioms are judged as more compliant than models that combine computational mechanisms in an ad hoc manner (Howes & Young, 1997).

Idioms are inherently pragmatic, indexed by the problems they solve (Chase & Simon, 1973). For example, consider the problem of representing overlapping receptive fields in connectionist networks. Hinton solved this problem in the mid-1970s by inventing COARSE CODING (Hinton interviewed in Anderson & Rosenfeld, 1998, p. 368). This is now recognized as the idiomatic solution to the more general problem of representing sparse information (Hinton, McClelland, & Rumelhart, 1986, p. 93). To take another example, the problem of representing a feature-value binding in connectionist networks can be solved by using the CONJUNCTIVE CODING idiom (Hinton et al., 1986, pp. 90–91), and the problem of representing multiple feature-value bindings can be solved by using the TEMPORAL SYNCHRONY idiom (Shastri & Ajjanagadde, 1993).

In the simplest case, an idiom can consist of a single computational mechanism. What is important is the problem the idiom solves or the functional role it serves, not its size. For example, connectionist architectures include additive units and sigma-pi units, which use different functions for computing the net input into a unit.

When designing a model, the choice between them is not a degree of freedom, but depends on the problem to be solved. Additive units implement a SIGNAL PROPAGATION idiom and sigma-pi units implement a SIGNAL GATING idiom (Rumelhart, Hinton, & McClelland, 1986, p. 73). Although SIGNAL PROPAGATION is more commonly used, some problems call for

SIGNAL GATING, for example, when implementing an attentional focus on the contents of working memory (Fix, Vitay, & Rougier, 2007).

4.2. Specificity and multiplicity

The relation between design problems, idioms, and cognitive architectures is rich and multiply determined. The same problem can be solved by different architectures using different idioms. For example, consider the problem of adding a content-addressable memory to a computational model. Connectionist architectures solve this problem using a DISTRIBUTED REPRESENTATIONS idiom whereby memory traces are encoded as patterns of activation over a set of fully interconnected units (Hinton et al., 1986, pp. 79–81). When part of the content is represented by clamping some of the units, the connections reconstruct the remainder of the content over the rest of the units. By contrast, the Soar production system architecture solves this problem using a PRODUCTIONS AS RECOGNITION MEMORY idiom. Productions can implement a content-addressable memory by encoding retrieval cues on their condition sides and memory traces on their action sides (Newell, 1990, pp. 165–166; Young & Lewis, 1999). Exemplar architectures solve this problem using different idioms. For example, TODAM2 stores item associations using the convolution and addition mechanisms of holography. Given part of the association, the remainder is retrieved using the correlation mechanism.

To take another example, consider the problem of imposing information processing limitations in cognitive architectures with parallel control structures. Soar models solve this problem by applying a DECLARATIVE MEMORY CONSTRAINTS idiom (Jones, Lebiere, & Crossman, 2007). To account for the fact that some syntactic constructions are more difficult to parse than others, NL-Soar includes a ''heads/dependents set'' that restricts the number of structural relations that can be considered for each constituent (Lewis, 1993; Young & Lewis, 1999). By contrast, connectionist models solve this problem by applying a FREQUENCY BY REGULARITY idiom. On this account, syntactic constructions are difficult to parse when they are less similar to the regular constructions that occur frequently in the linguistic environment, to which the network becomes attuned (MacDonald & Christiansen, 2002, p. 42).

The same computational mechanism can be used by different cognitive architectures to implement different idioms. For example, 4CAPS productions function like constraints on the activation levels of the declarative memory elements they match, implementing a CONSTRAINT SATISFACTION idiom (Just & Varma, 2002). By contrast, ACT-R productions function like goal-driven schemas conditioned on the contents of perceptual, motor, imaginal, and declarative memory buffers.

Finally, the same idiom can be implemented by different cognitive architectures using different computational mechanisms. As noted above, 4CAPS implements the CONSTRAINT SATISFACTION idiom using productions (Goldman & Varma, 1995). By contrast, IAC networks—a kind of connectionist architecture—implement this idiom using bidirectional connections between units (Kintsch, 1988; McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982).

4.3. Communication

Knowing the idioms of a cognitive architecture helps cognitive scientists design their own computational models; this is their pragmatic value. It also helps them understand computational models written by other members of the architectural community. In this regard, idioms serve a communicative function, increasing the intersubjective meaning of architectures. When inspecting a model written by another cognitive scientist, the tangles of computational mechanisms organize into idiomatic patterns signifying the problems that arose during model construction and how they were solved. The result is an understanding of the model's functional organization.

Idioms also increase the granularity of communication between cognitive scientists. The members of an architectural community know the same idioms—their functional roles and their implementations as patterns of computational mechanisms. When they communicate with one another, they do not need ''to build the field anew, starting from first principles and justifying the use of each concept introduced'' (Kuhn, 1996, pp. 19–20). Their discussions can use the succinct vocabulary of idioms rather than the verbose vocabulary of computational mechanisms.

4.4. Development and propagation

Another intersubjective role that idioms play is connecting members of architectural communities to their predecessors. Most problems that arise during the design of computational models are familiar, and thus they are solved by existing idioms. However, from time to time, a novel problem arises, one for which no appropriate idiom exists. This is particularly likely when a cognitive architecture is young and has only a small library of idioms, or when it is being applied to a new domain for the first time. When a novel problem arises, the cognitive scientist must construct a new pattern of computational mechanisms that solves it in a manner consistent with the architecture's worldview on human information processing. If the problem arises again and again, and if the pattern solves it in a satisfying way on most occasions, then it is on its way to idiomatic status. Other members of the architectural community will notice it, appreciate its pragmatic value, and apply it when the same problem crops up in their own modeling efforts. Over time, the pattern will join the architectural community's canon of idioms and will be taught to new members.

As an example of the development and propagation of an idiom, consider the problem of processing structured representations in connectionist architectures. Connectionists faced this problem in the late 1980s when they first began modeling high-level domains of cognition such as language comprehension. Elman (1990) solved it by using a novel pattern of recurrently connected units and reformulating the learning task to be sequential prediction, for example, predicting the next word in a sentence. He found that simple recurrent networks implicitly learned the structured representations necessary to make accurate sequential predictions. The community recognized this solution as more consistent with connectionist information processing than other proposals of the time (Hinton, 1990; Pollack, 1990; Smolensky, 1990), and it quickly spread. Today, RECURRENT CONNECTIONS is a familiar

idiom in the connectionist community, and members are connected to its originator when they use ''Elman nets'' in their own modeling efforts.

Another example is the development and propagation of the GOALS AS LTM ELEMENTS idiom in ACT-R. Earlier versions of this architecture included an infallible goal stack as a computational mechanism (Anderson, 1993; Anderson & Lebiere, 1998). As a result, earlier ACT-R models of problem solving could not account for the errors people make when they forget goals. Altmann and Trafton (2002) proposed that the goal stack was not a computational mechanism, but rather a pattern of computational mechanisms, where goals are encoded in and retrieved from long-term memory like other items, and therefore subject to decay and interference. Their model was able to account for errorful problem solving. Other ACT-R models found this pattern useful (Anderson & Douglas, 2001), and it is now accepted as an idiom within the ACT-R community (Taatgen & Anderson, 2010, p. 697).

5. The design of strange cognitive architectures

The conventional aesthetic criterion for the design and evaluation of cognitive architectures is parsimony. This section develops a complementary aesthetic criterion: Successful architectures are strange. The strangeness criterion has parallels in philosophies of art, for example, in Bacon's claim that ''[t]here is no excellent beauty that hath not some strangeness in the proportion.'' Just as artists strive to create revelatory artworks, cognitive scientists strive to create revelatory architectures. Strange architectures are provocative, often disturbing at first glance because they challenge prevailing assumptions. Cognitive scientists feel compelled to evaluate whether their surprising claims about human information processing are, in fact, true.

5.1. Noticing strangeness

When a strange architecture first appears, it will stand out against the field of accepted, conventional architectures (Von Restorff, 1933). The tension between the strange architecture and conventional architectures will unsettle cognitive scientists (Bruner & Postman, 1949), and their fears will grow to envelop the entire community (Kuhn, 1977). Should the strange architecture be ignored? Should it be evaluated like any other architecture, with the expectation that it will be quickly falsified? Or is it important—the leading edge of a new paradigm?

The ''logical response'' is to view a strange architecture as an ''error'': as a ''contradiction'' of conventional architectures. Most cognitive scientists will choose to ignore it, dismissing it as anomalous or unscientific (Kuhn, 1996; Mahoney, 1977; Mitroff, 1974). Others might conduct experiments designed to select between the strange architecture and conventional architectures. The expectation is that the former will be falsified, and summarily dismissed (Popper, 1963). However, some cognitive scientists will have a ''Hegelian response'': They will recognize that like a given conventional architecture (the *thesis*), the strange architecture (the *antithesis*) has value, and they will find a way to

reconcile the two (a *synthesis*). The logical response is usually appropriate—most new architectures are ill-conceived and quickly die well-deserved deaths. Therefore, on those rare occasions when a strange architecture merits further consideration, problems will arise.

As an example of the tension that strange architectures bring, consider the reaction to Grossberg's dissertation on connectionist networks when it first appeared at Stanford University in the mid-1960s. Rumelhart remembers that everyone:

> spent a great deal of time and effort trying to figure it out and failed completely… There were those who thought that, well, 'There is something very deep here, and it's beyond us. We just can't figure it out.' And there were those who thought that this was just a story, and it meant nothing, and we shouldn't pay any attention to it. (quoted in Anderson & Rosenfeld, 1998, p. 271)

Grossberg's connectionist networks arguably remain strange to this day to many cognitive scientists.

By contrast, when confronting strange architectures for the first time, some cognitive scientists will understand that the dissonance ''must mean something'' (Hinton quoted in Anderson & Rosenfeld, 1998, p. 375) and will see value in them. Their ''sense of the appropriate or the aesthetic'' (Kuhn, 1996, p. 155) will lead them to consider them further, in spite of their unorthodox computational mechanisms. This is illustrated by Newell's first exposure to the Pandemonium model in 1955, which he described as a conversion experience:

> [Selfridge and Dineen] had developed a mechanism that was so much richer than any other mechanism that I'd been exposed to that we'd entered another world as far as our ability to conceptualize. And that turned my life. (quoted in McCorduck, 1979, p. 134)

5.2. Accommodating strangeness

When a strange architecture has been deemed worthy of serious evaluation, the work has just begun, for now its computational consequences must be explored and its empirical scope delineated. This is a difficult and creative process because cognitive scientists ''must live in a world out of joint'' (Kuhn, 1996, p. 79). Returning to the previous example, after learning about Pandemonium, Newell set about marrying its parallel pattern-matching control structure with the heuristic search mechanisms he and Simon had developed (Simon, 1998, p. 70). The result was a number of strange architectures, including the General Problem Solver (GPS; Newell & Simon, 1961), production systems (Newell, 1973b; Newell & Simon, 1972), and Soar.

Another example of the design of strange architectures comes from DARPA's program on speech understanding systems in the 1970s. The two systems that came closest to meeting the program's ambitious goals were strikingly different. Hearsay-II was a

blackboard system, an incremental advance on the conventional production system architectures of the time (Erman, Hayes-Roth, Lesser, & Reddy, 1980, p. 218).[2] By contrast, Harpy was what we would today call a hidden Markov model (Lowerre & Reddy, 1980; Jurafsky & Martin, 2008, p. 332). Newell was an advisor on the Hearsay-II project. Although it would have been tempting to dismiss Harpy as a mere engineering success of no relevance to cognitive science, he instead attempted a Hegelian synthesis of the two systems. The result was HPSA77, a strange architecture that matches Hearsay-II-like productions (i.e., knowledge sources) using a Harpy-like network (Newell, 1980). This synthesis yielded new insights into cognition, such as the explanation of working memory constraints and serial processing as consequences of limitations on variable binding (Anderson, 1983, pp. 30–33; Touretzky & Hinton, 1988, p. 463).

5.3. Designing strange architectures

The strategies by which cognitive scientists design strange architectures can be partitioned into two classes. Data-driven strategies design them from without, in response to empirical regularities beyond the scope of conventional architectures. Formalism-driven strategies design them from within, through creative exploration with conventional architectures and computational formalisms drawn from outside cognitive science.

5.3.1. Data-driven strategies

5.3.1.1. Anomalous data: Many experiments in cognitive science are Popperian exercises, their possible outcomes predicted by computational models cast in conventional architectures. However, every once in a while, an experiment reveals a novel aspect of cognition—a previously unsuspected human ability or limitation—that is inconsistent with conventional accounts. When ''[c]onfronted with an anomaly,'' scientists are willing ''to try anything,'' and ''transition from normal to extraordinary research'' (Kuhn, 1996, p. 91). Some will design strange architectures that make the anomalous less so.

An example of the data-driven strategy dates to the dawn of the cognitive revolution. The early models of Newell and Simon depended on two characteristics of human cognition, the associative nature of memory, and the flexibility of problem solving (Simon, 1998, pp. 68–70). These characteristics were not easily expressed in the programming languages of the time (Simon quoted in Baars, 1986, pp. 364–365). To fill this gap, they designed the IPL family of programming languages (Laird & Rosenbloom, 1992, pp. 23–24; Simon, 1998, p. 68). These languages supported associative memory through linked list representations and operations for accessing elements by content rather than by location. These strange computational mechanisms proved critical for implementing the discrimination network at the heart of the EPAM model. The IPL languages supported flexible problem solving through another strange mechanism, dynamic memory allocation, which enabled the construction of new data structures at runtime. This mechanism was critical for implementing the LT and NSS Chess programs, which worked by progressively articulating problem spaces.

Another example of the data-driven strategy is the design of schema-based architectures in the mid-1970s. These strange architectures were spurred by empirical demonstrations that comprehension is ''constructive,'' a function of both linguistic input and prior knowledge (Bransford & Johnson, 1972). This contrasted with the then-dominant ''interpretive'' view, which focused exclusively on linguistic input. These demonstrations were initially dismissed—a logical rather than Hegelian response—but eventually proved too compelling to ignore (Jenkins quoted in Baars, 1986, pp. 243, 250–251). They prompted the design of strange architectures capable of supporting constructivist models of comprehension. These architectures organized prior knowledge using a variety of schematic representations: frames connected through inheritance mechanisms, scripts representing stereotypical events, and story schemas capturing generalized plot structures (Bobrow & Collins, 1975).

5.3.1.2. New experimental methods: It is often easy to dismiss an anomalous finding, however compelling, that defies explanation by conventional architectures. It is more difficult when a new experimental method generates an avalanche of anomalous findings. This more strongly compels the design of strange new architectures.

For example, consider the development of protocol analysis, where participants provide verbal reports of their thinking as they perform complex cognitive tasks such as problem solving and discourse comprehension (Ericsson & Simon, 1993; Pressley & Afflerbach, 1995). Newell remembers that:

> As soon as we got the protocols [of thinking studies run at RAND in 1955–6] they were fabulously interesting. They caught and just laid out a whole bunch of processes that were going on. (quoted in McCorduck, 1979, p. 212)

Protocol analysis differed from other methods of the time, such as tachistoscopic presentation of stimuli for brief periods, which were better suited for the study of simpler and more automatic forms of cognition. The flood of protocol data demanded the design of new architectures capable of accounting for the newly revealed complexities of cognition. In some cases, the data rather directly suggested the computational mechanisms the architectures needed to include.

> My recollection is that I just sort of drew GPS right out of subject 4 on problem D1 – all the mechanisms that show up in the book, the means-end analysis, and so on. (Newell quoted in McCorduck, 1979, p. 212)

GPS was strange compared to the mathematical models that dominated cognitive psychology at the time (e.g., Luce, Bush, & Galanter, 1963).

A more recent example is the rapid development of fMRI and other neuroimaging methods, which have resulted in an explosion of data on the neural bases of cognition. These data demand new architectural accounts. Some are meeting this demand by proliferating conventional architectures in strange ways. For example, the 3CAPS architecture, a production

system interpreter, focuses on how working memory resource limitations shape cognition (Just & Carpenter, 1992). Its successor, the 4CAPS architecture, understands cognition as the product of collaborative processing among brain areas. 4CAPS makes the strange design decision to model each brain area as an encapsulated 3CAPS production system whose resource limitations are re-interpreted as cortical in nature. Others are deconstructing conventional architectures in strange ways. For example, the latest version of ACT-R (Anderson, 2007) decomposes a conventional production system interpreter into its functional components (declarative memory, production matcher, etc.) and maps each to a different brain area (hippocampus and striatum, respectively).

5.3.2. Formalism-driven strategies

5.3.2.1. Emphasis: One way to design a strange architecture is to take a conventional computational mechanism and explore it, to see how much of cognition it can carry. Sometimes the answer is more than one would otherwise suspect, and the result is a strange architecture. For example, the classical Soar architecture (Newell, 1990) emphasizes its basic operation, the production rule, to a much greater degree than other production system architectures. Its claim that all long-term knowledge is encoded procedurally stands in contrast to ACT-R, which includes separate procedural and declarative memories. The classical Soar architecture is seemingly inconsistent with empirical evidence for separate procedural and declarative memory systems (Eichenbaum & Cohen, 2004).[3] Nevertheless, Soar models account for empirical regularities that are normally explained by appeal to declarative memory by employing productions in strange ways (Laird & Rosenbloom, 1992, p. 40; Young & Lewis, 1999). For example, the SCA model accounts for typicality effects, exemplar effects, and other properties of conceptual memory by encoding category exemplars as productions, and implementing exemplar retrieval using production matching (Miller & Laird, 1996).

The formalism-driven strategy of emphasis is often used in conjunction with the data-driven strategy of responding to anomalous data. When a conventional architecture is unable to account for an anomaly, one approach is to add a new computational mechanism that covers it. This has the undesirable effect of decreasing parsimony. Another approach is to follow Newell's advice that ''if you look at the architecture hard enough it will tell you how to solve that problem'' (quoted in Agre, 1993, p. 447). Sometimes an existing computational mechanism can be emphasized to an unusual degree to account for the anomaly in an unexpected way, transforming a conventional architecture into a strange one. For example, production system architectures face the problem of ''conflict resolution,'' or how to select which of multiple matching productions to fire next. In the 1970s, a number of control structures were proposed for solving this problem. The MEA conflict resolution scheme, for example, selects in a way that produces means-ends problem solving (Forgy, 1982). None of the proposed mechanisms proved to be of sufficient generality. Against this backdrop, Soar made the strange decision *not* to specify a fixed conflict resolution scheme. Instead, it emphasized other computational mechanisms in strange ways, using production rules and goals (in this case) in a recursive manner to decide which operator to apply next (Newell, 1990, pp. 170–174).

5.3.2.2. Combination: A cognitive architecture need ''not be a brand-new theory that replaces current work at every turn,'' but rather can ''put together and synthesize what we already know'' (Newell, 1990, p. 16). When computational mechanisms from different conventional architectures are juxtaposed, the result is often a strange architecture. This is the combination strategy. For example, the initial version of the ACT architecture, ACTE (Anderson, 1976), combined a declarative component (i.e., an associative network) adapted from the HAM model (Anderson & Bower, 1973) with a procedural component adapted from the PSG production system architecture (Newell, 1973b). This strange architecture differed from conventional production system architectures of the time, which lacked declarative long-term memories. [4]

The combination strategy has been used to design strange architectures that include both connectionist and exemplar mechanisms. One example is the construction-integration (CI) architecture (Kintsch, 1988). CI understands cognition as a sequence of two-phase cycles. During the construction phase, relevant knowledge is retrieved from long-term memory using computational mechanisms adapted from SAM, an exemplar model (Kintsch & Welsch, 1991; Raaijmakers & Shiffrin, 1981). During the integration phase, retrieved knowledge and incoming information are connected as nodes in a network, and a settling mechanism borrowed from IAC networks, a connectionist architecture, is used to identify central representations. Another example is ALCOVE (Kruschke, 1992), which combines exemplar mechanisms from the GCM (Medin & Schaffer, 1978; Nosofsky, 1984) with a connectionist error-driven learning mechanism (Rumelhart, Hinton, & Williams, 1986) to tune attention weights.

In the preceding examples, the combination strategy was applied by individual cognitive scientists. It can also be applied at the group level, by bringing together cognitive scientists from different architectural communities to mix and match their preferred computational mechanisms in novel ways. The result is sometimes strange architectures. For example, in the early 1980s, Carnegie Mellon University was a hotbed of symbolic architectures. Hinton was hired precisely because Newell was interested in ''getting a neural network presence'' (Hinton quoted in Anderson & Rosenfeld, 1998, p. 375). Hinton attempted several collaborations with symbolic cognitive scientists during his time there. Some were stillborn, as when Hinton and Newell tried to write a paper together about information processing and the brain, and failed (p. 375). However, other collaborations resulted in strange architectures, as when Hinton and Touretzky (who was then a symbolicist) designed a distributed connectionist production system (Touretzky & Hinton, 1988). This architecture made clear which symbolic mechanisms require localist implementations (e.g., productions) and which are amenable to distributed implementations (e.g., memory elements) (p. 463).

5.3.2.3. Importation: Most computational formalisms—mathematical theories, programming languages, and so on—have little to say about cognition. However, there are exceptions. When such a computational formalism is imported into cognitive science, the result is often a strange architecture. For example, the production system formalism was originally developed as a mathematical theory of computation on par with Turing machines (Post, 1943).

> The production system was one of those happy events, though in a minor key, that historians of science often talk about: a rather well-prepared formalism, sitting in wait for a scientific mission. (Newell & Simon, 1972, p. 889)

Newell and Simon imported this formalism into cognitive science, where it became a strange architecture for expressing models of high-level cognition. At roughly the same time, Chomsky (1957) imported this formalism into linguistics, as generative grammar—a strange framework for understanding language.

Another example of the importation strategy is the development of Boltzmann machines, a connectionist architecture that implements a constraint satisfaction style of information processing (Hinton & Sejnowski, 1986). Hinton and Sejnowski initially lacked a convergence proof—a guarantee that the network would settle into a unique solution. They solved this problem by importing the machinery of energy minimization from physics (Hinton quoted in Anderson & Rosenfeld, 1998, p. 372). Hinton and Sejnowski were also concerned about avoiding local minima during learning. They solved this problem by importing another notion from physics, that of simulated annealing (Sejnowski quoted in Anderson & Rosenfeld, 1998, pp. 322–323).

5.4. A visual metaphor

We conclude with a visual metaphor for understanding the two classes of strategies for designing strange architectures. Consider simple linear regression, where the goal is to identify the line that best accounts for (i.e., minimizes the squared deviations to) a set of points. Analogously, a goal of cognitive science is to identify the cognitive architecture that best accounts for a set of empirical regularities. If the regression line is mapped to a conventional architecture and the points are mapped to known empirical regularities, then Fig. 2A can be understood as representing the status quo in cognitive science.

There are two paths for progressing from conventional architectures to strange architectures. Data-driven strategies are spurred by the appearance of an anomalous datum, as shown in Fig. 2B. In regression terms, this corresponds to the appearance of an outlier. One method for handling outliers is to sample additional points in the empty region and to estimate a new line that accounts for the expanded set. This corresponds to the data-driven strategy of designing a new architecture that is strange (i.e., orthogonal) compared to the conventional architecture, and that accounts for the expanded data set; see Fig. 2D.

Formalism-driven strategies begin with the appearance of a strange architecture, one designed by emphasizing a computational mechanism of a conventional architecture or by combining the computational mechanisms of different conventional architectures in a novel manner or by importing a computational formalism from another discipline; this is shown in Fig. 2C. The new architecture is strange compared to the conventional architecture, or in the regression metaphor, the new line is orthogonal to the old line. Although the strange architecture does not provide a better account of existing empirical regularities, it spurs the collection of new empirical regularities. When the expanded data set is considered, the strange architecture provides a better overall account, as shown in Fig. 2D.

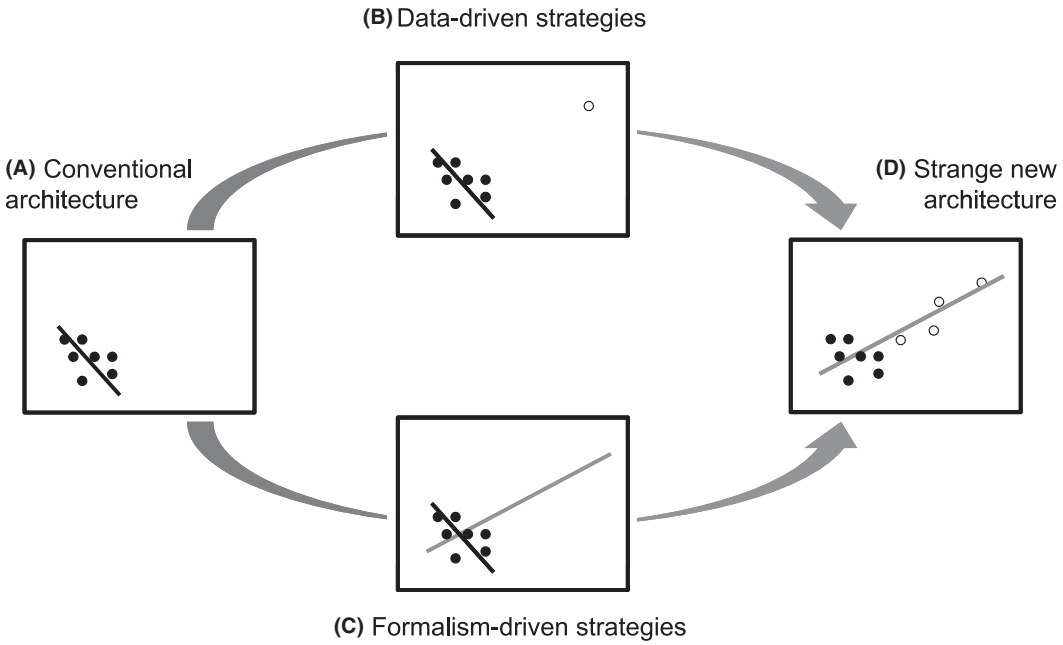

Fig. 2. (A) The *status quo*: A conventional architecture (black line) accounts for existing data (black dots). (B) An anomalous datum appears (white dot), prompting the design of a strange architecture. (C) A strange architecture (gray line) is designed, prompting the collection of new data. (D) The strange architecture provides the best account of the expanded data set.

6. Discussion

Empirical coverage and parsimony are the conventional criteria by which cognitive architectures and computational models are designed and evaluated. This paper has developed three additional criteria: Successful architectures possess subjective and intersubjective meaning, provide idioms that support the development of computational models, and make strange claims about human information processing that demand evaluation. These criteria were supported by examples from the history of cognitive science.

Empirical coverage, parsimony, subjective and intersubjective meaning, idioms, and strangeness are components of what might be called ''good taste'' in cognitive science theorizing. These criteria are not generally taught during graduate training, nor are they widely discussed in the literature. Rather, cognitive scientists must induce them for themselves through years of experience. These criteria are important, shaping the design of cognitive architectures and computational models and their evaluation through peer review, competitive argumentation, and empirical research. However, their tacit and idiosyncratic nature can lead to parochial judgments. What is in good taste for one cognitive scientist might be in bad taste for another.

An important goal for future research is to identify further criteria for the design and evaluation of cognitive architectures and computational models. Which criteria are supported by

strong historical records? Which are simply biases associated with particular architectural communities? Answering these questions cannot help but strengthen theorizing in cognitive science.

Notes

1. Howes and Young (1997) call these domain-specific mechanisms the ''model increment.''
2. Hearsay-II consists of a set of ''knowledge sources'' that collaborate to understand sentences by communicating their partial products via a central ''blackboard.'' Knowledge sources can be understood as condition-action pairs similar to production rules, although the condition and action aspects are implemented by arbitrary computer programs. The blackboard consists of a set of ''hypotheses'' that can be understood as items in shared declarative memory. In this regard, Hearsay-II and other blackboard architectures can be regarded as variants of production system architectures (Englemore & Morgan, 1988; Laird & Rosenbloom, 1992, p. 27). In fact, McCracken (1978) re-implemented Hearsay-II as a conventional production system.
3. Soar has recently been extended to include semantic and episodic declarative long-term memories (Laird, 2008).
4. The evolution of the ACT architecture continues to be driven by the combination strategy (Anderson, 2007, pp. 39–43).

Acknowledgments

I thank Wes Sherman, Jay Konopnicki, Mike Byrne, Nancy Nelson, Patricia Carpenter, Tim McNamara, Susan Goldman, John Bransford, David Noelle, Gordon Logan, Daniel Schwartz, and Lawrence Neeley for comments on prior versions of this paper, portions of which were presented at the 2006 AAAI Spring Symposium *What Went Wrong and Why: AI Research and Applications*. I thank John Anderson, Richard Young, Niels Taatgen, and two anonymous reviewers for comments on the current paper.

References

Agre, P. E. (1993). Interview with Allen Newell. *Artificial Intelligence*, *59*, 415–449.

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science*, *26*, 39–83.

Anderson, J. R. (1976). *Language, memory, and thought*. Hillsdale, NJ: Erlbaum.

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

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.

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

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford, England: Oxford University Press.

Anderson, J. R., & Bower, G. H. (1973). *Human associative memory*. Washington, DC: Winston & Sons.

Anderson, J. R., & Douglas, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *27*, 1331–1346.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Anderson, J. A., & Rosenfeld, E. (Eds.) (1998). *Talking nets: An oral history of neural networks*. Cambridge, MA: MIT Press.

Baars, B. J. (1986). *The cognitive revolution in psychology*. New York: The Guilford Press.

Bobrow, D. G., & Collins, A. (Eds.) (1975). *Representation and understanding: Studies in cognitive science*. New York: Academic Press.

Bransford, J. D., & Johnson, M. K. (1972). Contextual prerequisites for understanding: Some investigations of comprehension and recall. *Journal of Verbal Learning and Verbal Behavior*, *11*, 717–726.

Bruner, J. S., & Postman, L. (1949). On the perception of incongruity: A paradigm. *Journal of Personality*, *18*, 206–223.

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review*, *108*, 847–869.

Caplan, D., Baker, C., & Dehaut, F. (1985). Syntactic determinants of sentence comprehension in aphasia. *Cognition*, *21*, 117–175.

Carley, K., & Wendt, K. (1988). Electronic mail and scientific communication: A study of the Soar extended research group. *Knowledge: Creation, Diffusion, Utilization*, *12*, 406–440.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, *4*, 55–81.

Chomsky, N. (1957). *Syntactic structures*. The Hague, The Netherlands: Mouton.

Cohen, J. D., & Servan-Schreiber, D. (1992). Context, cortex and dopamine: A connectionist approach to behavior and biology in schizophrenia. *Psychological Review*, *99*, 45–77.

Cohen, N. J., & Squire, L. R. (1980). Preserved learning and retention of pattern-analyzing skill in amnesia: Dissociation of knowing how and knowing what. *Science*, *210*, 207–210.

Dougherty, M. R. P., Gettys, C. F., & Ogden, E. E. (1999). MINERVA-DM: A memory processes model for judgments of likelihood. *Psychological Review*, *106*, 180–209.

Eichenbaum, H., & Cohen, N. J. (2004). *From conditioning to conscious recollection: Memory systems of the brain*. Oxford, England: Oxford University Press.

Einstein, A., Podolsky, B., & Rosen, N. (1935). Can quantum-mechanical description of reality be considered complete? *Physical Review*, *47*, 777–780.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Elman, J., Bates, E., Karmiloff-Smith, A., Johnson, M., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness: Development in a connectionist perspective*. Cambridge, MA: MIT Press.

Englemore, R., & Morgan, T. (Eds.) (1988). *Blackboard systems*. Reading, MA: Addison-Wesley.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data (revised edition)*. Cambridge, MA: MIT Press.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. (1980). The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, *12*, 213–253.

Fix, J., Vitay, J., & Rougier, N. (2007). A distributed computational model of spatial memory anticipation during a visual search task. In M. V. Butz, O. Sigaud, G. Baldassarre, & G. Pezzulo (Eds.), *Anticipatory behavior in adaptive learning systems: From brains to individual and social behavior LNAI 4520* (pp. 170–188). Berlin: Springer.

Forgy, C. L. (1982). Rete: A fast algorithm for the many patterns/many objects match problem. *Artificial Intelligence*, *19*, 17–37.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns*. Reading, MA: Addison-Wesley.

Goldman, S. R., & Varma, S. (1995). CAPping the construction-integration model of discourse comprehension. In C. Weaver, S. Mannes, & C. Fletcher (Eds.), *Discourse comprehension: Essays in honor of Walter Kintsch* (pp. 337–358). Hillsdale, NJ: Erlbaum.

Goodman, N. (1968). *Ways of worldmaking*. Indianapolis, IN: Hackett.

Hanson, N. R. (1958). *Patterns of discovery*. Cambridge, England: Cambridge University Press.

Hinton, G. E. (1990). Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, *46*, 47–75.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed computing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 77–109). Cambridge, MA: MIT Press.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed computing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 282–317). Cambridge, MA: MIT Press.

Hintzman, D. L. (1986). Schema abstraction in a multiple-trace memory model. *Psychological Review*, *93*, 411–428.

Howes, A., & Young, R. M. (1997). The role of cognitive architecture in modeling the user: Soar's learning mechanism. *Human-Computer Interaction*, *12*, 311–343.

Jones, R. M., Lebiere, C., & Crossman, J. A. (2007). Comparing modeling idioms in ACT-R and Soar. In *Proceedings of the 8th International Conference on Cognitive Modeling* (pp. 49–54). Ann Arbor, MI.

Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing: An introduction to natural language processing, speech recognition, and computational linguistics* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, *99*, 122–149.

Just, M. A., & Varma, S. (2002). A hybrid architecture for working memory. *Psychological Review*, *109*, 54–64.

Just, M. A., & Varma, S. (2007). The organization of thinking: What functional brain imaging reveals about the neuroarchitecture of cognition. *Cognitive, Affective, and Behavioral Neuroscience*, *7*, 153–191.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, *95*, 163–182.

Kintsch, W., & Welsch, D. M. (1991). The construction-integration model: A framework for studying memory for text. In W. E. Hockley & S. Lewandowsky (Eds.), *Relating theory and data: Essays on human memory* (pp. 367–385). Hillsdale, NJ: Erlbaum.

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

Kuhn, T. S. (1977). The essential tension. *Selected studies in scientific tradition and change*. Chicago: University of Chicago Press.

Kuhn, T. S. (1996). *The structure of scientific revolutions* (3rd ed.). Chicago: The University of Chicago Press.

Laird, J. E. (2008). Extending the Soar cognitive architecture. In P. Wang, B. Goertzel, & S. Franklin (Eds.), *Proceedings of the First Conference on Artificial General Intelligence* (pp. 224–235). Memphis, TN.

Laird, J. E., & Rosenbloom, P. S. (1992, Winter). The research of Allen Newell. *AI Magazine*, *13*, 17–45.

Laird, J. E., & Rosenbloom, P. S. (1996). The evolution of the Soar cognitive architecture. In D. M. Steier & T. M. Mitchell (Eds.), *Mind matters: A tribute to Allen Newell* (pp. 1–50). Mahwah, NJ: Erlbaum.

Lewis, R. L. (1993). *An architecturally-based theory of human sentence comprehension*. Unpublished Ph.D. dissertation, Pittsburgh, PA: Carnegie Mellon University.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 375–419.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*, 492–527.

Lovett, M. C. (2005). A strategy-based interpretation of Stroop. *Cognitive Science*, *29*, 493–524.

Lowerre, B., & Reddy, D. R. (1980). The HARPY speech understanding system. In W. Lea (Ed.), *Trends in speech recognition* (pp. 576–586). Englewood Cliffs, NJ: Prentice-Hall.

Luce, R. D., Bush, R. R., & Galanter, E. (Eds.) (1963). *Handbook of mathematical psychology* (Volumes I-III). New York: Wiley.

MacDonald, M. C., & Christiansen, M. H. (2002). Reassessing working memory: A comment on Just & Carpenter (1992) and Waters & Caplan (1996). *Psychological Review*, *109*, 35–54.

Mahoney, M. J. (1977). Publication prejudices: An experimental study of confirmatory bias in the peer review system. *Cognitive Therapy and Research*, *1*, 161–175.

Marr, D. (1982). *Vision: A computational investigation into the human representation and presentation of visual information*. New York: W. H. Freeman.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*, 375–407.

McClelland, J. L., & Rumelhart, D. E. (1988). *Explorations in parallel distributed processing*. Cambridge, MA: MIT Press.

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

McCracken, D. L. (1978). *A production system version of the Hearsay-II speech understanding system*. Unpublished Ph.D. dissertation, Pittsburgh, PA: Carnegie Mellon University.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*, 207–238.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, *104*, 3–65.

Miller, C. S., & Laird, J. E. (1996). Accounting for graded performance within a discrete search framework. *Cognitive Science*, *20*, 499–537.

Mitroff, I. I. (1974). Norms and counter-norms in a select group of the Apollo moon scientists: A case study of the ambivalence of scientists. *American Sociological Review*, *38*, 579–595.

Mozer, M., Smolensky, P., Touretzky, D., Elman, J., & Weigand, A. (Eds.) (1994). *Proceedings of the 1993 connectionist models summer school*. Hillsdale, NJ: Erlbaum.

Neurath, O., Carnap, R., & Morris, C. (Eds.) (1955). *Foundations of the unity of science, volume 1*. Chicago: University of Chicago Press.

Newell, A. (1973a). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual information processing* (pp. 283–308). New York: Academic Press.

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

Newell, A. (1980). Harpy, production systems, and human cognition. In R. Cole (Ed.), *Perception and production of fluent speech* (pp. 289–395). Hillsdale, NJ: Erlbaum.

Newell, A. (1989). Putting it all together. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A* (pp. 399–440). Hillsdale, NJ: Erlbaum.

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

Newell, A., & Simon, H. A. (1961). Computer simulation of human thinking. *Science*, *134*, 2011–2017.

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

Nosofsky, R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*, 104–114.

Oppenheim, P., & Putnam, H. (1958). Unity of science as a working hypothesis. In H. Feigl, M. Scriven, & G. Maxwell (Eds.), *Minnesota studies in the philosophy of science, volume II: Concepts, theories, and the mind-body problem* (pp. 3–36). Minneapolis: University of Minnesota Press.

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. Cambridge, MA: MIT Press.

Plaut, D. C., & Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology*, *10*, 377–500.

Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence*, *36*, 77–105.

Popper, K. R. (1963). *Conjectures and refutations: The growth of scientific knowledge*. New York: Harper & Row.

Post, E. L. (1943). Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, *65*, 197–215.

Pressley, M., & Afflerbach, P. (1995). *Verbal protocols of reading: The nature of constructively responsive reading*. Hillsdale, NJ: Erlbaum.

Pylyshyn, Z. W. (1984). *Computation and cognition: Toward a foundation for cognitive science*. Cambridge, MA: MIT Press.

Raaijmakers, J. G. W., & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review*, *88*, 93–134.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. In D. E. Rumelhart & J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed computing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 45–76). Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, *89*, 60–94.

Rumelhart, D. E., & McClelland, J. L. (1986). PDP models and general issues in cognitive science. In D. E. Rumelhart & J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed computing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 110–146). Cambridge, MA: MIT Press.

Shastri, L., & Ajjanagadde, V. (1993). From simple associations so systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, *16*, 417–494.

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.

Simon, H. A. (1998). Allen Newell: 1927–1992. *IEEE Annals of the History of Computing*, *20*, 63–76.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, *46*, 159–216.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, *29*, 1213–1228.

Taatgen, N. A., & Anderson, J. R. (2010). The past, present, and future of cognitive architectures. *Topics in Cognitive Science*, *2*, 693–704.

Touretzky, D. S., & Hinton, G. E. (1988). A distributed connectionist production system. *Cognitive Science*, *12*, 423–466.

Townsend, J. T. (1974). Issues and models concerning the processing of a finite number of inputs. In B. H. Kantowitz (Ed.), *Human information processing: Tutorials in performance and cognition* (pp. 133–185). Hillsdale, NJ: Erlbaum.

Von Restorff, H. (1933). Uber die wirkung von bereichsbildungen im spurenfeld [The effects of field formation in the trace field]. *Psychologie Forschung*, *18*, 299–334.

Warrington, E. K., & Shallice, T. (1984). Category-specific semantic impairments. *Brain*, *107*, 829–853.

Wittgenstein, L. (1958). *Philosophical investigations* (G. E. M. Anscombe, Trans.). Englewood Cliffs, NJ: Prentice-Hall.

Young, R. M., & Lewis, R. L. (1999). The Soar cognitive architecture and human working memory. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 224–256). Cambridge: Cambridge University Press.