Cognitive Theory, SOAR

Richard L. Lewis

Department of Computer & Information Science

The Ohio State University

2015 Neil Avenue

Columbus, OH 43210

DRAFT: October 29, 1999, 10:12 AM

145 Cognitive Theory, SOAR

SOAR is a computational theory of human cognition that takes the form of a general cognitive architecture (Laird, Newell, & Rosenbloom, 1987; Newell, 1990; Rosenbloom, Laird, & Newell, 1992). SOAR (not an acronym) is a major exemplar of the *architectural approach* to cognition, which attempts the unification of a range of cognitive phenomena with a single set of mechanisms, and addresses a number of significant methodological and theoretical issues common to all computational cognitive theories (Anderson & Lebiere, 1998; Newell, 1990; Pylyshyn, 1984). SOAR is also characterized by a set of specific theoretical commitments shaped primarily by attempting to satisfy the *functional* requirements for supporting human-level intelligence, manifest in SOAR's parallel existence as a state-of-the art artificial intelligence system (Laird et al., 1987). This focus on functionality, and its attendant theoretical commitments, is what makes SOAR both distinctive and controversial in cognitive psychology. SOAR represents the last major work of Allen Newell, one of the founders of modern cognitive science and artificial intelligence, and a pioneer in the development of architectures as a class of cognitive theory.

1. Multiple constraints on mind and computational theories of cognition

Newell (1980a; 1990) described the human mind as a solution to a set of functional constraints (e.g., exhibit adaptive (goal-oriented) behavior, use language, operate with a body of many degrees of freedom) and a set of constraints on construction (a neural system, grown by embryological processes, arising through evolution). The structure of SOAR is shaped primarily by three of the functional constraints: (a) exhibiting flexible, goal-driven behavior, (b) learning continuously from experience, and (c) exhibiting real-time cognition (elementary cognitive behavior must be evident within about a second).

The emergence of computational models of cognition in information processing psychology (and artificial intelligence) represented a significant theoretical advance by providing the first proposals for physical systems that could, in principle, satisfy the functional constraints of exhibiting intelligence (Newell, Shaw, & Simon, 1958; Newell & Simon, 1972). However, they raised a set of difficult methodological and theoretical issues that cognitive science still grapples with today. Among these issues are: (a) the problem of *irrelevant specification* (in a complex computer program, which of the myriad aspects of the program carry theoretical content, and which are irrelevant implementation details?) (Reitman, 1965); (b) the problem of *too many degrees of freedom* (an unconstrained computer program can be modified to fit any data pattern); and (c) the problem of *identifiability* (any sufficiently general proposal for processing schemes or representations can mimic the input/output characteristics of any other general processing or representation scheme (Anderson, 1978; Pylyshyn, 1973).

2. SOAR as a confluence of five major technical ideas in cognitive science

SOAR can be seen as a confluence of five major technical ideas in cognitive science, which, taken together, are intended to address the three functional constraints summarized above, as well as the fundamental methodological issues concerning computational models.

2.1 Physical symbol systems

SOAR is a *physical symbol system*. *The physical symbol system hypothesis* asserts that physical symbol systems are the only class of systems that can in principle satisfy the constraint of supporting intelligent, flexible behavior. Physical symbol systems are a reformulation of Turing universal computation (Church, 1936; Turing, 1936) that identifies *symbol processing* as a key feature of intelligent computation. The requirement is that the system be capable of manipulating and composing symbols and symbol structures—physical patterns with associated processes that give the patterns the power to denote either external entities or other internal symbol structures (Newell, 1980a; Newell, 1990; Simon, 1996). The key to the universality of Turing machines (and physical symbol systems) is their *programmability*: content can be added to the systems (in the form of programs) to change their behavior, yielding indefinitely many response functions.

2.2 Cognitive architectures

SOAR is a *cognitive architecture.* A cognitive architecture is a theory about the fixed computational structure of cognition (Anderson & Lebiere, 1998; Newell, 1990; Pylyshyn, 1984). Computational systems that are programmable must have some kind of fixed structure that processes the variable content: a set of primitive processes, memories, and control structures. The theoretical status of this underlying structure has not always been clear in cognitive models. For example, when a cognitive model is programmed in Lisp, the theorist intends to make some theoretical claims about the program (e.g., that the steps of the program corresponds in some way to the cognitive steps of the human performing the task), but probably intends to make no theoretical claims about Lisp as the architecture that executes the program (e.g., the fact that unused memory structure are reclaimed via a garbage collection process is theoretically irrelevant).

A cognitive architecture explicitly specifies a fixed set of processes, memories, and control structures that are capable of encoding content and executing programs. Cognitive models for specific tasks can be developed in such architectures by programming them. The theoretical status of various parts of a programmed implementation is now considerably clarified: what counts is the structure of the architecture (not its particular implementation), and the cognitive model's program, which makes a set of specific commitments about the form and content of knowledge used in a specific task. Thus, implemented cognitive architectures go a long way toward solving the irrelevant specification problem.

Cognitive architectures, especially those with temporal mappings and integrated learning mechanisms, can also address the degrees of freedom problem and identifiability problem in four ways. First, to the extent that architectures have a constrained temporal mapping, the space of possible programs that yield both the required functionality and temporal profile is considerably reduced. Second, to the extent that architectures have learning components that can acquire new knowledge (e.g. about a specific task), the form of that knowledge is no longer freely under control of the theorist. Third, to the extent that architectures are programmable (and are also constrained by a temporal mapping or learning mechanism), they permit a single set of processing assumptions to be applied to a diverse range of tasks, constraining that theory by a broader range of data. Fourth, to the extent that cognitive architectures are comprehensive and include some perceptual and motor components, they can be used to provide closed-loop models of *complete* tasks, so that no explanatory power need be ascribed to anything external to the model.

2.3 Production systems

All long term memory in SOAR is held in the form of *productions* (Anderson, 1993; Newell, 1973). Each production is a condition-action pair. The conditions form access paths and the actions form the memory contents. Productions continuously match against a declarative working memory that contains the momentary task context, and matching productions put their contents (actions) back into the working memory. Productions are the lowest level of elementary memory access available in SOAR, and Newell's (1990) temporal mapping onto human cognition places them approximately at the 10ms level. This mapping provides strong constraints on the shape of cognitive models built in SOAR that must operate in real time.

SOAR's productions form a recognition memory. Such recognition memories have a number of features that make them attractive as models of human memory: they are *associational* in nature (access is via the contents of working memory); they are *fine-grained and independent* (which makes them a good match for continuous, incremental learning mechanisms); they are *dynamic* (a production system by itself defines a computationally complete system that can yield behavior; other processes are not needed to access or execute the memory structures); and they are *cognitively impenetrable* (their contents and structure may not be arbitrarily searched over, examined, or modified, but only accessed via automatic association). All of these properties place them in sharp contrast to memories in digital computers, which are static structures (not processes), freely addressable by location.

2.4 Search in problem spaces supported by a two-level automatic/deliberate control structure

SOAR achieves all cognition by *search in problem spaces*, and architecturally supports this by a flexible, two-level *recognize-decide-act* control structure. Problem spaces are based in part on the idea that search in combinatoric spaces is the fundamental process for attainment of difficult tasks. The nature of such search is seen most easily in tasks like chess that have a well-defined set of *operators* and *states*. A search space consists of a set of (generated) representational states and operators that transition between states.

Problem spaces as realized in SOAR extend the standard notion of search in an important direction: problems spaces are taken to be the fundamental way that humans accomplish *all* cognitive tasks, including routine (i.e, well-practiced) tasks. SOAR is therefore one realization of the *problem space hypothesis* (Newell, 1980b), which asserts that all deliberate cognitive activity occurs in problem spaces. The key to this move lies in the role of knowledge in problem spaces: problem spaces freely admit of any amount of knowledge for guiding search, executing operators, or formulating the space initially in response to a task. Because SOAR provides a set of mechanisms (described next) that support this kind of knowledge use, behavior in SOAR spans the well-known continuum between knowledge-intensive processing (little search) and knowledge-lean processing (much search) (Newell, 1990).

Supporting knowledge-driven search places strong functional demands on the architecture's control structure: at any step in the problem solving process—selecting the next operator, generating the next state, etc.—any relevant knowledge must be brought to bear. There are two parts to the solution to this problem: the mechanisms for appropriate indexing of the knowledge, and the mechanisms for retrieving and applying the relevant knowledge during search. The indexing concerns learning, discussed below.

For retrieving and applying the knowledge during search, SOAR relies on a two-level control structure that separates the automatic access of knowledge via the productions from the deliberate level of problem solving. Each cognitive step is accomplished by a *recognize-decide-act* cycle. In the recognize phase, all productions that match the current state fire, producing new content in the working memory. Part of this retrieved content is about what the system should do next—the possible operators to try in the current state, the relative desirability of these operators (e.g., operator A is better than operator B), and so on. Next, in the *decide* phase, a fixed (domain independent) decision procedure sorts out these preferences in working memory to determine if they converge on a consistent decision. In the event that this processing clearly determines the next step, the decision procedure places in working memory an assertion about what that step should be. In the *act* phase, that step is taken (by additional production rule firings): the move to the next state in internal problem space search, or the release of motor intentions in external interaction. If it is not clear what to do next (e.g., several operators have been proposed, but no knowledge is evoked to prefer one option to another, or there are conflicts in the retrieved knowledge), an *impasse* has arisen, and the decision procedure records in working memory the type of the impasse, and sets a subgoal of resolving that impasse. In this way, SOAR's problem solving gives rise automatically to a cascade of subgoals whenever the knowledge delivered by the recognition memory is insufficient for the current task.

The critical feature of this control structure is its *run-time, least-commitment* nature: each local decision in the problem space is made at execution time by assembling whatever relevant bits of knowledge can be retrieved (by automatic match) at that moment. Decisions are not fixed in advance, and there are no architectural barriers to the kinds of knowledge that can be brought to bear on the decisions.

2.5 Continuous, impasse-driven learning

SOAR continuously acquires new knowledge in its long term memory through an experience-based learning mechanism called *chunking* (Laird et al., 1987; Rosenbloom & Newell, 1986). This mechanism generates new productions in the long term memory by preserving the results of problem solving that occurred in response to impasses. The conditions of the new production consist of aspects of the working memory state just before the impasse, and the actions of the production consist of the new knowledge that resolved the impasse (for example, an assertion that one of the proposed operators is to be preferred to the other in the current situation). Upon encountering a similar situation in the future, the production will automatically match and retrieve the knowledge that permits SOAR to avoid the impasse. Thus, chunking is a mechanism that converts problem solving into recognition memory, continuously moving SOAR from knowledge-lean to knowledge-rich processing.

Chunking in SOAR has two important functional properties. First, it begins to provide a solution to the knowledge-indexing problem raised earlier. The system assembles its own indices out of the contents of working memory in a way that is directly aimed at making the knowledge retrievable when it is relevant to the immediate demands of the task at hand. Second, learning permeates all aspects of cognition in SOAR. Chunking applies to all kinds of impasses, so any problem space function is open to learning improvements: problem space formulation, operator generation, operator selection ,and so on.

3. Major architectural implications and specific domains of application

SOAR can be used as a theory in multiple ways (Newell, 1990). Qualitative predictions can be drawn from SOAR as a verbal theory, without actually running detailed computer simulations. These qualitative predictions can be both domain-general (cutting across all varieties of cognitive behavior) and domain-specific. The theory can be also be applied to specific domains by developing detailed computational models of a task; this involves programming SOAR by adding domain-specific production rules to its long-term memory, and generating behavioral traces.

3.1 Domain-independent predictions

A principal prediction of a theory of human cognition is that humans are intelligent; the only way to clearly make that prediction is to demonstrate it operationally. SOAR makes this prediction only to the extent that the system has been demonstrated to exhibit intelligent behavior. As a state of the art AI system that has been applied to difficult tasks (ranging from algorithm design to scheduling problems), SOAR makes the prediction to a greater degree than other psychological theories.

SOAR makes a number of general predictions related to long-term memory and skill (Newell, 1990). These include the prediction that procedural skill transfer is essentially by identical elements, and will usually be highly specific (Singley & Anderson, 1989; Thorndike, 1903); the bias of *Einstellung* will occur—the preservation of learned skill when it is no longer useful (Luchins, 1942); the encoding specificity principle (Tulving, 1983) holds; and recall will generally take place by a generate-and-recognize process (Kintsch, 1970). The best-known of SOAR's general predictions is the *power law of practice*, which relates the time to do a task to the number of times the task has been performed (Newell & Rosenbloom, 1981; Snoddy, 1926).

3.2 Domain-specific predictions

SOAR models have been constructed across a range of task domains, and the behavior of the models has been compared to human data on those tasks. One area that has received considerable attention is human-computer interaction (HCI). Some of the successes in this area, such as a detailed model of transcription typing (John, 1988), are a result of SOAR inheriting the results of the GOMS theory (Goal, Operators, Methods, and Selection rules), a theory developed in HCI to predict the time it takes expert users to do routine tasks (Card, Moran, & Newell, 1983). (GOMS can be seen at one level as a specialization of SOAR, missing features such as learning and impassing). Other SOAR HCI models depend crucially on SOAR's real-time interruptability (a function of the two-level control structure) and SOAR's learning mechanism. SOAR models have been developed of real-time interaction and learning in video games (John, Vera, & Newell, 1994), novice-to-expert transitions in computer menu navigation (Howes & Young, 1997), and a programmer's interaction with a text editor (Altmann & John, 1999), among others.

SOAR models have also been developed of problem solving (Newell, 1990), sentence processing (Lewis, 1997), concept acquisition (Miller & Laird, 1996), and interaction with educational microworlds (Miller, Lehman, & Koedinger, 1999). In all SOAR models (as with any cognitive model), the explanatory power is shared to varying degrees by both the content posited by the theorist for the particular task and the architectural mechanisms. For example, in the sentence processing model, SOAR's control structure and learning mechanism, coupled with the real-time constraint, lead directly to a theory of ambiguity resolution that yields a novel explanation of apparent modularity effects and their malleability (Lewis, 1996a; Newell, 1990), but the architecture provides little apparent constraint on the choice of grammatical theory, which also play a role in the empirical predictions (Lewis, 1996b). Similarly, the general theory of episodic indexing of attention events embodied in the text editor model depends critically on SOAR's continuous chunking mechanism (Altmann & John, 1999), while the specific behavioral traces are a function, in part, of task strategies that could be accommodated by alternative architectures.

4. Critiques of SOAR, and future directions

Critiques of SOAR fall into three major classes: critiques of specific models built within SOAR, critiques of the architecture itself, and critiques of the general methodological approach of building comprehensive architectural theories. For example, specific empirical critiques have been made of SOAR models of the Sternberg memory search task (Lewandowsky, 1992) and immediate reaction tasks (Cooper & Shallice, 1995). The theoretical challenge is understanding the extent to which the empirical problems can be resolved within the existing architecture, or whether they point back to problems in the architecture itself (Newell, 1992b). (The fact that the latter is a real possibility demonstrates that the architectural approach has made some headway on the identifiability and degrees of freedom problems.)

At the architectural level, nearly every major assumption of SOAR has been challenged in literature (see the multiple book review in BBS for a range of assessments (Newell, 1992a)). Many of these architectural-level criticisms have been aimed at the *uniformity* assumptions in SOAR (all tasks as problem spaces, all long-term memory as productions, all learning as chunking), which appears at first to run strikingly against the prevailing mode of theorizing in both cognitive psychology and cognitive neuroscience, which emphasizes functional specialization and distinctions over computational generality. The evaluation of SOAR in light of these concerns is not always transparent, however. For example, the analysis of SOAR's implications for modularity (particularly in language processing) revealed that SOAR is not only consonant with, but even predicts, many of Fodor's diagnostics of modular systems (Lewis, 1996a; Newell, 1990).

Finally, the general approach to cognitive theory that SOAR embraces has come under sharp criticism (most notably by Cooper and Shallice, 1995) for not living up to the promise of addressing the methodological concerns identified above, and for not yielding theories with deep empirical coverage that clearly gain their explanatory power from general architectural mechanisms. To the extent that these critiques depend on practice with the SOAR theory specifically, their implications for the broader approach are insecure. Other architectural theories (e.g., ACT (Anderson & Lebiere, 1998) and EPIC (Meyer et al., 1995)) exist in the field, and each has adopted somewhat different ways of dealing with these methodological issues that may or may not make them suspect to the same criticisms.

The evolution of SOAR as a theory, and its broader role in cognitive science, is likely to proceed along two fronts. First, SOAR will remain an important source of ideas for developing theories of complex cognition, even for those theorists who do not embrace the architecture whole cloth, or reject the architectural methodology. A harbinger of this can be seen in cognitive neuroscience: as researchers begin to tackle the problem of understanding the nature of "executive" processes and their realization in the brain, models like SOAR can provide concrete proposals for a set of functionally sufficient mechanisms for the control of deliberate cognition; (see the recent volume on working memory and executive control for evidence of such interaction (Miyake & Shah, 1999)). Second, SOAR will continue to evolve as a unified set of mechanisms itself, informed in part by the continued application of SOAR to difficult AI problems, and in part by the continued construction and empirical evaluation of detailed models of cognitive tasks that focus on unique aspects of the architecture.

See also: Power law of learning; Cognitive theory, ACT; Architectures of cognition; Production systems in cognitive psychology; Artificial intelligence in cognitive science

Altmann, E. M., & John, B. E. (1999). Episodic indexing: A model of memory for attention events. *Cognitive Scienceq, 23*(2), 117-156.

Anderson, J., & Lebiere, C. (1998). *Atomic Components of Thought*: Lawrence Erlbaum.

Anderson, J. R. (1978). Arguments concerning representations for mental imagery. *Psychological Review, 85*(4), 249-277.

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

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Church, A. (1936). An unsolvable problem of elementary number theory. *The American Journal of Mathematics, 58,* 345-363.

Cooper, R., & Shallice, T. (1995). SOAR and the case for unified theories of cognition. *Cognition, 55*(2), 115-149.

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

John, B. E. (1988). *Contributions to engineering models of human-computer interaction.* , Carnegie Mellon University.

John, B. E., Vera, A. H., & Newell, A. (1994). Toward real-time GOMS: A model of expert behavior in a highly interactive task. *Behavior and Information Technology, 13*(255-267).

Kintsch, W. (1970). Models for free recall and recognition. In D. A. Norman (Ed.), *Models of Human Memory* . New York: Academic Press.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33,* 1-64.

Lewandowsky, S. (1992). Unified cognitive theory: Having one's apple pie and eating it. *Behavioral and Brain Sciences, 15*(3), 449-450.

Lewis, R. L. (1996a). Architecture Matters: What SOAR has to say about modularity. In D. a. M. Steier, T. (Ed.), *Mind Matters: Contributions to Cognitive and Computer Science in Honor of Allen Newell* . Hillsdale, NJ: Erlbaum.

Lewis, R. L. (1996b). Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research*, *25*(1), 93--115.

Lewis, R. L. (1997). Specifying architectures for language processing: Process, control, and memory in parsing and interpretation. In M. a. P. Crocker, Matt and Clifton, Charles (Ed.), *Architectures and Mechanisms for Language Processing* . Cambridge: Cambridge University Press.

Luchins, A. S. (1942). Mechanization in problem solving. *Psychological Monographs, 54*(6).

Meyer, D. E., Kieras, D. E., Lauber, E., Schumacher, E. H., Glass, J., Zurbriggen, E., Gmeindl, L., & Apfelblat, D. (1995). *Adaptive Executive Control: Flexible Human Multiple-task Performance without pervasive immutable response-selection bottlenecks* : University of Michigan, Department of Psychology.

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

Miller, C. S., Lehman, J. F., & Koedinger, K. R. (1999). Goals and learning in microworlds. *Cognitive Science*, *23*(3), 305-336.

Miyake, A., & Shah, P. (Eds.). (1999). *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*. Cambridge: Cambridge University Press.

Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual Information Processing* . New York: Academic Press.

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

Newell, A. (1980b). Reasoning, problem solving and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.), *Attention and Performance VIII* . Hillsdale, NJ: Erlbaum.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, Massachusetts: Harvard University Press.

Newell, A. (1992a). Precis of Unified Theories of Cognition. *Behavioral and Brain Sciences, 15*(3), 425-492.

Newell, A. (1992b). SOAR as a unified theory of cognition: Issues and explanations. *Behavioral and Brain Sciences, 15*(3), 464-492.

Newell, A., & Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive Skills and Their Acquisition* . Hillsdale, NJ: Erlbaum.

Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review, 65*, 151-166.

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

Pylyshyn, Z. W. (1973). What the mind's eye tells the mind's brain: a critique of mental imagery. *Psychological Bulletin, 80*(1), 1-24.

Pylyshyn, Z. W. (1984). *Computation and Cognition*. Cambridge, MA: Bradford / MIT Press.

Reitman, W. (1965). *Cognition and Thought*. New York: Wiley.

Rosenbloom, P. S., Laird, J. E., & Newell, A. (Eds.). (1992). *The SOAR Papers: Research on Integrated Intelligence*. Cambridge, MA: MIT Press.

Rosenbloom, P. S., & Newell, A. (1986). The chunking of goal hierarchies: A generalized model of practice. In R. S. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach II* . Los Altos, California: Morgan Kaufman.

Simon, H. A. (1996). The patterned matter that is mind. In D. a. M. Steier, T. (Ed.), *Mind Matters: Contributions to Cognitive and Computer Science in Honor of Allen Newell* . Hillsdale, NJ: Erlbaum.

Singley, M. K., & Anderson, J. R. (1989). *The Transfer of Cognitive Skill*. Cambridge, MA: Harvard University Press.

Snoddy, G. S. (1926). Learning and stability. *Journal of Applied Psychology, 20*, 1-36.

Thorndike, E. L. (1903). *Eeducational Psychology*. New York: Lemke & Buechner.

Tulving, E. (1983). *Elements of Episodic Memory*. New York: Oxford University Press.

Turing, A. M. (1936). *On computable numbers, with an application to the Entscheidungsproblem*. Paper presented at the Proceedings of the London Mathematics Society.

Richard L Lewis
Department of Computer and Information Science and Center for Cognitive Science
The Ohio State University