A Cognitive Odyssey:
From the Power Law of Practice to a General Learning Mechanism and Beyond

Paul S. Rosenbloom
University of Southern California

This article traces a line of research that began with the establishment of a pervasive regularity in human performance – the Power Law of Practice – and proceeded through several decades' worth of investigations that this opened up into learning and cognitive architecture. The results touch on both cognitive psychology and artificial intelligence, and more specifically on the possibility of building general learning mechanisms/systems. It is a story whose final chapter is still to be written.

In the fall of 1979 I returned to my position as a PhD student in the Computer Science Department at Carnegie Mellon University (CMU) after a year as a visiting graduate student in the Psychology Department at the University of California, San Diego (UCSD). I had taken a leave of absence from CMU in 1978 with a strong interest in both artificial intelligence and cognitive psychology – viewing them as complementary ways of understanding the processes underling intelligent behavior – but in training and thought processes I was very much a computer scientist rather than a psychologist. During my year at UCSD in the LNR Research Group (the acronym refers to the group's three founders, Peter Lindsay, Donald Norman and David Rumelhart, but at this time it was led by the latter two along with James –Jay- McClelland) I had the privilege to witness the beginning of the resurgence of neural network modeling in psychology, while also learning something about what it

meant to be a psychologist.

On returning to CMU, I brought with me some ideas I had developed in the area of activation-based production systems, along with an implemented system – eXperimental Activation Based Production System (XAPS) – based on combining my earlier experiences at CMU working with Allen Newell on production system architectures with what I had recently learned about neural networks at UCSD. XAPS will come back into the story later on, but on returning to CMU I did not have a strong sense of where to go from there with it, and was somewhat at loose ends overall in figuring out what to do next.

Around this same time, Allen Newell had agreed to contribute to the Sixteenth Annual Carnegie Symposium on Cognition, to be held in May of 1980 on the acquisition of cognitive skills (Anderson, 1981). He wanted to look more deeply into the purported power law structure of human practice curves, and into what the implications of this might be for the overall architecture of cognition. Cognitive architecture – the fixed structure underlying cognition – was an abiding interest for Allen Newell (culminating in his final book on *Unified Theories of Cognition*, Newell, 1990); and it was the opportunity to work with him on this topic that originally intrigued me enough as a prospective graduate student (in 1976) to lure me from California to Pittsburgh.

What was particularly intriguing about the power law structure of practice curves in this context was that if this

regularity were truly as ubiquitous as it was beginning to appear to be – and if it weren't dictated by either the structure of the environment or some simple statistical regularity (as is the case with the bell curve and Zipf's law, Zipf, 1949) – it would likely reveal a ubiquitous constraint on how learning was implemented and functioned as part of the architecture of human cognition. This notion became even more compelling later on as it became clear that the leading models of practice in existence did not lead to power law learning curves.

So, in 1979, Newell asked me to begin a new effort with him on this topic, pointing towards the upcoming Symposium, and I agreed. The first major outcome of this effort was the analysis – via a combination of regression and search – of a broad range of practice curves, focusing particularly on data relating time to perform a task versus the number of times the task had been performed, that showed such curves were best modeled via a general power law of the form $T = A + B(N + E)^{-\alpha}$; where T is the time to perform the task, N is the number of times that the task has been performed during the experiment (i.e., the number of trials), α is the learning rate, A is the asymptotic response time, E is a parameter reflecting the amount of effective practice that has occurred prior to the start of the experiment, and B is the performance time (above asymptote) the first time the task is performed (i.e., when N+E = 1).

This first outcome helped establish the ubiquity of the Power Law of Practice, and set the stage for the second major outcome: the analysis of existing models of practice to determine whether any of them could yield power law practice curves – with the conclusion that they couldn't – along with the development of a new model. This new model – the Chunking Theory of Learning – was grounded in the earlier concept of chunks from work in perception and memory (Miller, 1956), and proved capable of producing practice curves that closely mimicked power laws by enabling performance to be increasingly based on higher level patterns that short circuited lower level processing. Together, these two outcomes formed the basis for the article being celebrated in this special issue (Newell & Rosenbloom, 1981).

Of the two outcomes – establishing the Power Law of Practice and developing the Chunking Theory of Learning – it was the second one that had the far bigger impact on Newell's and my research careers. It led us, and an increasing set of collaborators – most notably, John Laird, with whom Newell and I jointly formed and led the long-term, multi-disciplinary Soar project (Rosenbloom, Laird & Newell, 1993) – on more than two decades worth of investigations into the role of chunking in the architecture of cognition and of the implications of the interactions between chunking and the other aspects of cognition – both architecture and content (i.e., knowledge represented on top of the architecture) – for cognitive processing in general. (Although Allen Newell died in 1992, and I moved on to other things in 1998, efforts have continued to this day under John Laird and others.)

The first steps beyond the paper being celebrated here involved instantiating the abstract conceptual model provided by the Chunking Theory via a sequence of implementations of chunking in the context of cognitive architectures, in which experimental tasks could be performed, performance time could be measured, and chunking could speed up performance over repeated trials through the acquisition of rules that bypassed the more extensive processing that was previously required to perform the tasks. Each successive implementation involved either a significant generalization of chunking or of the underlying cognitive architecture within which it was implemented.

The first implementation was task dependent (Rosenbloom & Newell, 1982; Rosenbloom & Newell, 1987). The cognitive architecture – XAPS2, a descendent of the XAPS architecture mentioned earlier that was adapted to the constraints imposed by the needs of chunking – was task independent. However, the implementation of chunking only worked for one task: a 1023-choice reaction-time task involving ten lights and ten buttons (Seibel, 1963). The implementation was hardwired to chunk regions of lights and regions of buttons and to relate light regions to button regions, and thus wasn't task independent and couldn't be considered truly architectural. This implementation produced learning curves, but it was too slow to run for many trials, and chunking learned too quickly to acquire plausible learning curves. A meta-simulation enabled "simulating the simulator" at a more rapid pace, and adding a probability of chunk acquisition slowed down the effective rate of learning, to where a power law learning curve could be generated for this task.

The second implementation yielded a task-independent version of chunking (Rosenbloom, 1986; Rosenbloom & Newell, 1988). The core innovation was conceiving of performance in terms of a very fine-grained goal hierarchy, and defining chunks in terms of patterns of goal inputs, patterns of goal results, and connections relating input and result patterns. Chunks sped up task performance by replacing extended goal processing with firing of rules that generated results from inputs. In this model, tasks were represented as specific goal hierarchies, but the processing and chunking of these hierarchies could be defined in a task independent manner, providing the first truly architectural

implementation of chunking. This drove several additional changes to the underlying cognitive architecture, to introduce goal processing into the architecture and to eliminate activation (which turned out to hide information from chunking, leading to incorrect learning), yielding the XAPS3 architecture. Practice curves were generated for the Seibel task as well as for a range of stimulus-response compatibility tasks (Duncan, 1977; Fitts and Seeger, 1953; Morin and Forrin, 1962), demonstrating the task-independent nature of chunking and enabling the interactions between practice and compatibility to be explored.

The third implementation involved a radical generalization of the underlying cognitive architecture. John Laird, a fellow graduate student who had simultaneously been working with Newell on a general, rule-based, goal-oriented, problem solving architecture called Soar (Laird, 1986), created an implementation of chunking in the context of Soar. Soar was based on a problem space model of performance, consisting of states, and operators that generate new states from existing states. It embodied the concept of Universal Subgoaling, which enabled goals to be generated automatically not only for task-oriented activities but also for arbitrary problems in the system's own performance, such as the inability to choose between a set of candidate actions to perform or the lack of a problem space within which to work on task performance. This enabled chunking to improve performance on both task and reflective/meta-level goals (Rosenbloom, Laird & Newell, 1988).

The implementation of chunking in Soar yielded a system that not only could produce power law practice curves (Rosenbloom, Laird & Newell, 1989), but also had much broader possibilities as a model of human cognition, as well as showing significant promise as a general architecture for artificial intelligence (Laird, Rosenbloom & Newell, 1986a; Laird, Newell & Rosenbloom, 1987). This led to an explosion of work investigating different tasks and capabilities in Soar, with learning by chunking being a key aspect of much of this work. Several early overviews of the varieties of research on learning in Soar can be found in Rosenbloom, et al. (1986), Laird & Rosenbloom, 1987 and Steier, et al. (1987); and a later, albeit briefer, overview can be found along with many of the papers from 1991 and earlier in Rosenbloom, Laird & Newell, 1993.

We were emboldened by this early success with chunking in Soar, and by a sense of nearly limitless untapped potential, to set out to test the hypothesis that a general problem solver in combination with a simple experience-based learning mechanism – that is, chunking, which would feed off of the generality of knowledge and

processing of the problem solver – could together yield a general learning mechanism capable of improving performance on all tasks, basing its improvements on any knowledge that is available, improving all aspects of the system, and transferring learned knowledge from one situation to other comparable situations (Laird, Rosenbloom & Newell, 1986a). The remainder of this article focuses on the history of this expedition.

The core of the early work in Soar focused on its use as a general problem solver, in which weak-method – that is, knowledge-lean – search in problem spaces, along with Universal Subgoaling, were the key performance concepts, and puzzles and games formed the main task domains (Laird & Newell, 1983). Thus, the first results with chunking in Soar focused primarily on improving task performance through the acquisition of search-control rules and macro-operators – by chunking over subgoals in which lookahead search yielded evaluation and preference information useful in deciding which options to take – that would reduce the amount of search that was initially required for these tasks (Laird, Rosenbloom & Newell, 1984; Laird, Rosenbloom & Newell, 1986a). It was shown that such learned knowledge could improve performance within trial (i.e., during the trial in which the knowledge is learned), across trial (i.e., on later instances of the same task), and across task (i.e., to different but related tasks).

Around this time we also began to stretch Soar's reach beyond puzzles and games to more substantive knowledge-intensive tasks – such as the classic R1, computer configuration, expert system problem (McDermott, 1982) – with the accompanying question of what chunking would do in such problems. What we found was that the knowledge in such domains could be partitioned into task definition knowledge, which specifies a set of problem spaces within which solutions can be found, and search control knowledge that helps avoid missteps while leading the system directly to solutions. Without the former knowledge, problems can't be solved – or even defined – while without the latter knowledge problems can be solved but at the cost of extensive search. One main result was that when the system was initialized with just the task definition knowledge, chunking could perform a form of knowledge compilation by acquiring the control knowledge that converted a search-intensive system into a knowledge-intensive one (Rosenbloom, et al., 1985). This work was then followed up with investigations of learning in other expert systems domains, such as algorithm design (Steier, 1987; Steier & Newell, 1988), chemical process design (Modi & Westerberg, 1989), and medical diagnosis (Washington & Rosenbloom, 1993).

Although, through such investigations chunking was

beginning to yield a broadly useful skill acquisition mechanism for artificially intelligent systems, two key problems had started to show up in the performance of the resulting learned systems. The first problem was overgeneralization – whenever a chunk failed to capture some of the conditions under which it was appropriate to be used, it could apply where it should not, leading to errors in performance (Laird, Rosenbloom & Newell, 1986b). Several sources of overgeneralization were identified, some of which were correctable through changes to the code that implemented chunking, while more fundamental sources had to be dealt with by learning more chunks that enabled the system to recover from the effects of the incorrect ones that had been learned earlier (Laird, 1988; Pearson & Laird, 2005).

The second problem was that the system would slow down as more rules were learned. From a cognitive modeling perspective, this could perhaps be ignored as an implementation detail, but from both a pragmatic and an AI perspective, it couldn't be ignored as it could cause the system to slow down to the point of being unusable. It turned out that this utility problem (Minton, 1990) had two aspects: (1) some individual chunks were expensive, in that the time to determine their applicability was exponential in their size (Tambe, Newell & Rosenbloom, 1990); and (2) even when no individual chunks were expensive, adding thousands or millions of chunks to the system could significantly slow down processing (Doorenbos, Tambe & Newell, 1992). Approaches to the first aspect included bounding the expressibility of chunks (and of all rules used in the system) to limit their match costs (Tambe, Newell & Rosenbloom, 1990; Tambe & Rosenbloom, 1994) and modifying chunking so that the cost of using a chunk was bounded by the cost of the problem solving from which it was learned (Kim & Rosenbloom, 2000). The second aspect was resolved via optimizations to the rule matcher, resulting in successful experiments in which over a million chunks were learned without a substantial slow down (Doorenbos, 1994).

However, beyond these issues a deeper problem lurked. Although chunking originated in the context of research on perception and memory – as a core element in the memorization of perceived patterns (Miller, 1956; Chase & Simon, 1973) – it was becoming increasingly apparent that in Soar it was far from obvious how chunking could be used to support even the simplest form of rote learning. Consider, for example, the problem of learning to recall a newly perceived object. What is desired to be learned is a new rule that can generate the perceived object later, perhaps in the presence of relevant cues. However, if the object is attended to, and examined piece by piece in a subgoal so as to generate it as the result of the subgoal – as is necessary for it to be generated by a learned chunk – the resulting chunk will test all aspects of the object as well as generate them, thus only being applicable when the object is already available. Such a rule can speed up the action of copying the object when it is perceived – which is what was essentially done in the subgoal – but cannot recall it when it is not already present.

One approach to this problem might have been to add a new learning mechanism to Soar in support of rote learning, as for example exists in ACT-R (Anderson & Lebiere, 1998). However, continuing to test the hypothesis that chunking in Soar could provide the basis for a general learning mechanism demanded persevering to see if there were a solution that didn't require a new architectural learning mechanism, and what the implications of such a solution might be should it exist. Although this was easier said than done, a solution to the conundrum was eventually worked out based on several new concepts. First, it was necessary to reorganize the problem solving in the subgoal over which recall learning was to occur so as to reconstruct the newly perceived object from what was already known rather than from what was currently perceived, so that generation of the new object by the resulting chunk wouldn't first test for its being perceived (Rosenbloom, Laird & Newell, 1987). Second, the fact that a chunk existed to generate an object had to be taken as evidence that the object had been perceived at some point in the past (Rosenbloom & Aasman, 1990). This second idea was needed so that it would be possible to distinguish between all of the things that could be reconstructed via what was known – many of which would have corresponded to things that never were or never could be seen (such as a pink elephant) – and those things that were actually perceived (and encoded into recall chunks).

The details of all of this are yet subtler than what has been presented here, including the need to acquire and use recognition chunks for testing the success of the generation process and the role of search control in chunk acquisition. However, one of the most interesting take away lessons from it was that by trying to live within the constraints of the architecture, rather than jumping immediately to add new mechanisms when the existing ones don't obviously do what is desired, we were driven to a model of rote learning that was inherently reconstructive, with interesting parallels to generate-recognize theories of recall, such as Watkins & Gardiner (1979). Over the years, this general approach to what we ended up referring to as data chunking, to distinguish it from the earlier skill-acquisition uses of chunking, was extended beyond simple recall (and recognition) to cover cued recall (Rosenbloom, Laird &

Newell, 1989) and concept acquisition from multiple examples (Rosenbloom & Aasman, 1990; Miller & Laird, 1991; Miller & Laird, 1996; Rogers, 1996); and to a proposal for a general framework for acquiring new knowledge (Rosenbloom, 1988).

Simultaneously with the investigation of data chunking, and continuing on for years afterwards, work was also proceeding on a variety of other aspects, uses, and implications of chunking in Soar. For example, we looked at utilizing advice from humans in guiding problem solving towards what would be useful to learn (Golding, Rosenbloom & Laird, 1987; Laird et al., 1990); acquiring new knowledge, including the task definition knowledge delineating entire problem spaces, from instruction (Yost & Newell, 1988; Lewis, Newell & Polk, 1989; Huffman & Laird, 1995); learning to draw valid conclusions in immediate reasoning tasks such as syllogistic reasoning (Polk & Newell, 1988; Polk, Newell & Lewis, 1989); abbreviating strategic processing in the Tower of Hanoi (Ruiz & Newell, 1989); acquiring generalized rules and abstract plans by chunking over abstracted problem spaces (Unruh & Rosenbloom, 1989); learning to interact robotically with external environments (Laird & Rosenbloom, 1990; Laird et al., 1991); learning number conservation knowledge (Simon, Newell & Klahr, 1991); learning to integrate multiple sources of knowledge for natural language processing (Lehman, Lewis & Newell, 1991); producing and overcoming Einstellung/masking effects (Tambe & Rosenbloom, 1993); learning to explain the system's own behavior (Johnson, 1994); learning from observation of the performance of others (van Lent & Laird, 1999; van Lent & Laird, 2001); and extending the procedural knowledge of a system in a complex, dynamic environment (Pearson & Laird, 2005).

The most sophisticated learning system based on chunking in Soar to date combined work on learning through instruction (Huffman & Laird, 1995) and error-recovery/procedural-learning in dynamic domains (Pearson & Laird, 2005). The combined system could learn how to perform new tasks through instruction, recover from errors in performance by learning new procedural knowledge, and even use instruction to help in the error recovery process (Laird, Pearson & Huffman, 1997). The most comprehensive examination of Soar as a model of human memory, learning and skill can be found – along with a broader examination of Soar as a candidate Unified Theory of Cognition – in Newell (1990).

Chunking at its essence moves performance along a store-versus-compute – or, equivalently, space-versus-time, deliberation-versus-reaction or controlled-versus-automatic (Shiffrin & Schneider, 1977) – tradeoff, extracting lessons from experience in executing tasks and storing those lessons

as rules that influence later behavior without the need of repeating the earlier processing. As such, it bears a family resemblance to a range of other learning and compilation mechanisms, such as memo functions (Michie, 1968), macro-operator learning (Fikes, Hart & Nilsson, 1972; Korf, 1985), production composition/compilation (Lewis, 1978; Anderson & Lebiere, 1998), explanation-based learning (Mitchell, Keller & Kedar-Cabelli, 1986; Rosenbloom & Laird, 1986), knowledge compilation (Laird, Rosenbloom & Newell, 1986b; Selman & Kautz, 1996), and partial evaluation (Jones, Gomard & Sestoft, 1993). A big part of what has made it interesting – and something it shares with production compilation in ACT-R (Anderson & Lebiere, 1998) – is its close connection with a more broadly functional cognitive architecture. This combination has proven to yield an intriguing range of capabilities and behaviors.

Yet, there are also three new investigations of learning in Soar that start from the notion that, although chunking has shown itself to have quite broad applicability, there are still key forms of learning that are either too hard when using chunking as the only learning mechanism, or whose non-architectural/content components disrupt or compete with critical task-based processing when the learning occurs via chunking over the kinds of learning-oriented problem solving required by data chunking. These three efforts are investigating augmenting Soar with new learning mechanisms that are particularly valuable in its current primary role as an architecture for intelligent agents and virtual humans in simulations and games (Tambe et al., 1995; Hill et al., 1997; Jones et al., 1999; Laird, 2002; Hill et al., 2003; Wray et al., 2005); in particular, to support reinforcement learning (Nason & Laird, 2005), episodic learning (Nuxoll & Laird, 2004), and semantic memory/learning. This represents a focused backing off from the initial hypothesis of chunking in Soar providing a completely general learning mechanism, but it is a well-informed backing off, based on years of investigation into the initial hypothesis. It will be interesting to see how the newly augmented Soar architecture will function, and how these new mechanisms will work with each other and with the already existing ones.

To review the path we have traversed in this article, and in this line of research, we began with establishing a general regularity in human performance – the Power Law of Practice – that was then used as a constraint in proposing a learning mechanism to be incorporated as part of an overall cognitive architecture. A sequence of implementations ultimately yielded chunking in Soar, along with the associated hypothesis that this combination could provide a general learning mechanism. A wide range of loosely coordinated investigations into this hypothesis turned up

much evidence in its favor, as well as some substantial issues. Some of these issues have led to the current focus of learning research in Soar on augmenting chunking with three new learning mechanisms, with intended applications in the areas of intelligent agents and virtual humans. This overall path has led us a long way from our starting point in analyzing human practice curves, but it hopefully shows the fundamental power of sustained efforts focusing on regularities in human performance and their interactions with cognitive architectures.

References

Anderson, J. R. (1981). *Cognitive Skills and their Acquisition*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

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

Doorenbos, R. (1994). Combining left and right unlinking for matching a large number of learned rules. *Proceedings of the Twelfth National Conference on Artificial Intelligence* (pp. 451-458). AAAI.

Doorenbos, R., Tambe, T., and Newell, A. (1992). Learning 10,000 chunks: What's it like out there? *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp 830-836). Menlo Park, CA: AAAI Press/The MIT Press.

Duncan, J. (1977). Response selection rules in spatial choice reaction tasks. In S. Dornic (Ed.), *Attention and Performance VI*. Hillsdale, NJ: Erlbaum.

Fikes, R. E., Hart, P. E. & Nilsson, N. J. (1972). Learning and executing general robot plans. *Artificial Intelligence*, 3, 251-288.

Fitts, P. M. & Seeger, C. M. (1953). S-R compatibility: Spatial characteristics of stimulus and response codes. *Journal of Experimental Psychology*, 48, 483-492.

Golding, A. R., Rosenbloom, P. S. & Laird, J. E. (1987). Learning general search control from outside guidance, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp. 334-337). Milan, Italy: IJCAII.

Hill, R. W., Chen, J., Gratch, J., Rosenbloom, P. S. & Tambe, M. (1997). Intelligent agents for the synthetic battlefield: A company of rotary wing aircraft, Proceedings, *Ninth Conference on Innovative Applications of Artificial Intelligence* (pp. 1006-1012). AAAI.

Hill, R., Gratch, J., Marsella, S., Rickel, J., Swartout, W., & Traum, D. (2003). Virtual Humans in the Mission Rehearsal Exercise System. *Künstliche Intelligenz (KI Journal)*. Special issue on Embodied Conversational Agents.

Huffman, S. B., Laird, J. E. (1995). Flexibly instructable agents. *Journal of Artificial Intelligence Research*, 3, 271-324.

Johnson, W. L. (1994). Agents that learn to explain themselves. *Proceedings of the Twelfth National Conference on Artificial Intelligence* (pp. 1257-1263). AAAI.

Jones, N. D., Gomard, C. K. & Sestoft, P. (1993). *Partial Evaluation and Automatic Program Generation*. Prentice Hall International.

Jones, R. M., Laird, J. E., Nielsen P. E., Coulter, K., Kenny, P., & Koss, F. (1999). Automated intelligent pilots for combat flight simulation, *AI Magazine*, 20, 27-42.

Kim, J. & Rosenbloom, P. S. (2000). Bounding the cost of learned rules. *Artificial Intelligence*, 120, 43-80.

Korf, R. E. (1985). Macro-operators: A weak method for learning. *Artificial Intelligence*, 26, 35-77.

Laird, J. E. (1986). Universal subgoaling. In J. E. Laird, P. S. Rosenbloom & A. Newell (Eds.), *Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies* (pp. 3-131). Hingham, MA: Kluwer Academic Publishers.

Laird, J. E. (1988). Recovery from incorrect knowledge in Soar. *Proceedings of the National Conference on Artificial Intelligence* (pp. 618-623).

Laird, J. E. (2002). Research in human-level AI using computer games. *Communications of the ACM*, 45, 32-35.

Laird, J. E., Hucka, M., Yager, E. S., Tuck, C. M. (1990). Correcting and extending domain knowledge using outside guidance. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 235-243).

Laird, J. E., Hucka, M., Yager, E. S., Tuck, C. M. (1991). Robo-Soar: An integration of external interaction, planning and learning using Soar. In W. Van de Velde (Ed.), *Toward Learning Robots*. Boston, MA: MIT Press.

Laird, J. E. & Newell, A. (1983). A Universal Weak Method. Tech. report #83-141, Carnegie Mellon University Computer Science Department. (Republished in P. S. Rosenbloom, J. E. Laird, Newell, A. (1993). *The Soar Papers: Research on Integrated Intelligence* (pp. 245-292). Cambridge, MA: MIT Press.)

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

Laird, J. E., Pearson, D. J. & Huffman, S. B. (1997). Knowledge-directed adaptation in multi-level agents. *Journal of Intelligent Information Systems*, 9, 261-275.

Laird, J. E. & Rosenbloom, P. S. (1987). Research on learning in Soar, *Proceedings of the Second Annual Artificial Intelligence Research Forum* (pp. 240-253). Palo Alto, CA: NASA Ames Research Center.

Laird, J. E. & Rosenbloom, P. S. (1990). Integrating execution, planning, and learning in Soar for external environments, *Proceedings of the Eighth National*

Conference on Artificial Intelligence (pp. 1022-1029). Boston, MA: MIT Press.

Laird, J. E., Rosenbloom, P. S. & Newell, A. (1984). Towards chunking as a general learning mechanism, *Proceedings of the National Conference on Artificial Intelligence* (pp. 188-192). Austin, TX: AAAI.

Laird, J. E., Rosenbloom, P. S. & Newell, A. (1986a). Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning*, 1, 11-46.

Laird, J. E., Rosenbloom, P. S. & Newell, A. (1986b). Overgeneralization during knowledge compilation in Soar. In T. G. Dieterich (Ed.), *Proceedings of the Workshop on Knowledge Compilation* (pp. 46-57). Otter Crest, OR: AAAI.

Lehman, J. F., Lewis, R. L. & Newell, A. (1991). Integrating knowledge sources in language comprehension. *Proceedings of the Thirteenth Annual Meeting of the Cognitive Science Society* (pp. 461-466). Hillsdale, NJ: Erlbaum.

Lewis, C. H. (1978). *Production System Models of Practice Effects*. Ph.D. Dissertation, University of Michigan.

Lewis, R. L., Newell, A. & Polk, T. A. (1989). Toward a Soar theory of taking instructions for immediate reasoning tasks. *Proceedings of the 11th Annual Conference of the Cognitive Science Society* (pp. 514-521).

McDermott, J. (1982). R1: A rule-based configurer of computer systems. *Artificial Intelligence*, 19, 39-88.

Michie, D. (1968). "Memo" functions and machine learning. *Nature*, 218, 19-22.

Miller, G. A. (1956). The magic number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81-97.

Miller, C. S., Laird, J. E. (1991). A constraint-motivated lexical acquisition model. *Proceedings of the 13th Annual Conference of the Cognitive Science Society* (pp. 827-831).

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

Minton, S. (1990). Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence*, 42, 363-392.

Mitchell, T. M., Keller, R. M. & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, 1, 47-80.

Modi, A. K. & Westerberg, A. W. (1989) Integrating learning and problem solving with a chemical process designer. Presented at the Annual Meeting of American Institute of Chemical Engineers. (Republished in P. S. Rosenbloom, J. E. Laird, Newell, A. (1993). *The Soar Papers: Research on Integrated Intelligence* (pp. 727-753). Cambridge, MA: MIT Press.)

Morin, R. E. & Forrin, B. (1962). Mixing two types of S-R associations in a choice reaction time task. *Journal of Experimental Psychology*, 64, 137-141.

Nason, S. & Laird, J. E. (2005). Soar-RL: integrating reinforcement learning with Soar. *Cognitive Systems Research*, 6, 51-59.

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

Newell, A. & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive Skills and their Acquisition* (pp. 1-55). Hillsdale, NJ: Erlbaum.

Nuxoll, A. & Laird, J. E. (2004). A cognitive model of episodic memory integrated with a general cognitive architecture. *Proceedings of the Sixth International Conference on Cognitive Modeling* (pp. 220-225).

Pearson, D. J . & Laird, J. E. (2005). Incremental learning of procedural planning knowledge in challenging environments. *Computational Intelligence*, 21, 414-439.

Polk, T. A. & Newell, A. (1988). Modeling human syllogistic reasoning in Soar. *Proceedings of the 10th Annual Conference of the Cognitive Science Society* (pp. 181-187).

Polk, T. A., Newell, A. & Lewis, R. L. (1989). Toward a unified theory of immediate reasoning in Soar. *Proceedings of the 11th Annual Conference of the Cognitive Science Society* (pp. 506-513).

Rogers, S. (1996). Symbolic performance & learning in complex environments. *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (p. 1405). AAAI.

Rosenbloom, P. S. (1986). The chunking of goal hierarchies: A model of practice and stimulus response compatibility. In J. E. Laird, P. S. Rosenbloom & A. Newell (Eds.), *Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies* (pp. 135-282). Hingham, MA: Kluwer Academic Publishers.

Rosenbloom, P. S. (1988). Beyond generalization as search: Towards a unified framework for the acquisition of new knowledge. In G. F. DeJong (Ed.), *Proceedings of the AAAI Symposium on Explanation-Based Learning* (pp. 17-21). Stanford, CA: AAAI.

Rosenbloom, P. S. & Aasman, J. (1990). Knowledge level and inductive uses of chunking (EBL), *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 821-827). Boston, MA: MIT Press.

Rosenbloom, P. S. & Laird, J. E. (1986). Mapping explanation-based generalization onto Soar, *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 561-567). Philadelphia, PA: AAAI.

Rosenbloom, P. S., Laird, J. E., McDermott, J., Newell, A. & Orciuch, E. (1985). R1-Soar: An experiment in

knowledge-intensive programming in a problem-solving architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7, 561-569.

Rosenbloom, P. S., Laird, J. E. & Newell, A. (1987). Knowledge level learning in Soar, *Proceedings of Sixth National Conference on Artificial Intelligence* (pp. 499-504). AAAI.

Rosenbloom, P. S., Laird, J. E. & Newell, A. (1988). Meta-levels in Soar. In P. Maes & D. Nardi (Eds.), *Meta-Level Architectures and Reflection* (pp. 227-240). Amsterdam, Netherlands: North Holland.

Rosenbloom, P. S., Laird, J. E. & Newell, A. (1989). The chunking of skill and knowledge. In B. A. G. Elsendoorn & H. Bouma (Eds.), *Working Models of Human Perception* (pp. 391-410). London, England: Academic Press.

Rosenbloom, P. S., Laird, J. E. & Newell, A. (Eds.). (1993). *The Soar Papers: Research on Integrated Intelligence*. Cambridge, MA: MIT Press.

Rosenbloom, P. S., Laird, J. E., Newell, A., Golding, A. & Unruh, A. (1986). Current research on learning in Soar. In T. M. Mitchell, J. G. Carbonell & R. S. Michalski (Eds.), *Machine Learning: A Guide to Current Research* (pp. 281-290). Boston, MA: Kluwer Academic Press.

Rosenbloom, P. S. & Newell, A. (1982). Learning by chunking: Summary of a task and a model, *Proceedings of the National Conference on Artificial Intelligence* (pp. 255-257). Pittsburgh, PA: AAAI.

Rosenbloom, P. S. & Newell, A. (1987). Learning by chunking: A production-system model of practice. In D. Klahr, P. Langley & R. Neches (Eds.), *Production System Models of Learning and Development* (pp. 221-286). Cambridge, MA: Bradford Books/MIT Press.

Rosenbloom, P. S. & Newell, A. (1988). An integrated computational model of stimulus-response compatibility and practice. In G. H. Bower (Ed.), *The Psychology of Learning and Motivation*, Volume 21 (pp. 1-52). San Diego, CA: Academic Press.

Ruiz, D. & Newell, A. (1989). Tower-noticing triggers strategy-change in the Tower of Hanoi: A Soar model. *Proceedings of the Annual Conference of the Cognitive Science Society* (pp. 522-529).

Seibel, R. (1963). Discrimination reaction time for a 1,023-alternative task. *Journal of Experimental Psychology*, 66, 215-226.

Selman, B. & Kautz, H. (1996). Knowledge compilation and theory approximation. *Journal of the ACM*, 43, 193-224

Shiffrin, R. M. & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review*, 84, 127-190.

Simon, T., Newell, A. & Klahr, D. (1991). A computational account of children's learning about number conservation. In D. H. Fisher, M. J. Pazzani & P. Langley (Eds.), *Concept Formation: Knowledge & Experience in Unsupervised Learning*. Los Altos, CA: Morgan Kaufmann.

Steier, D. M. (1987). Cypress-Soar: A case study in search and learning in algorithm design, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp. 327-330). IJCAII.

Steier, D. M., Laird, J. E., Newell, A., Rosenbloom, P. S., Flynn, R., Golding, A., Polk, T. A., Shivers, O. G., Unruh, A. & Yost, G. R. (1987). Varieties of Learning in Soar: 1987. In P. Langley (Ed.), *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 300-311). Los Altos, CA: Morgan Kaufmann Publishers, Inc.

Steier, D. M. & Newell, A. (1988). Integrating multiple sources of knowledge into Designer-Soar, an automatic algorithm designer. *Proceedings of the Seventh National Conference on Artificial Intelligence* (pp. 8-13). AAAI.

Tambe, M., Johnson, W. L., Jones, R. M., Koss, F., Laird, J. E., Rosenbloom, P. S. & Schwamb, K. B. (1995). Intelligent agents for interactive simulation environments. *AI Magazine*, 16, 15-39.

Tambe, M., Newell, A. & Rosenbloom, P. S. (1990). The problem of expensive chunks and its solution by restricting expressiveness. *Machine Learning*, 5, 299-348.

Tambe, M. & Rosenbloom, P. S. (1993). On the masking effect, *Proceedings of the Eleventh National Conference on Artificial Intelligence* (pp. 526-533). Washington, D.C.: AAAI.

Tambe, M. & Rosenbloom, P. S. (1994). Investigating production system representations for non-combinatorial match. *Artificial Intelligence*, 68, 155-199.

Unruh, A. & Rosenbloom, P. S. (1989). Abstraction in problem solving and learning, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 681-687). Detroit, MI: IJCAII.

van Lent, M., & Laird, J. E. (1999). Learning hierarchical performance knowledge by observation. *Proceedings of the International Conference on Machine Learning*.

van Lent, M., Laird, J. E. (2001). Learning procedural knowledge through observation. *Proceedings of the First International Conference on Knowledge Capture*.

Washington, R. & Rosenbloom, P. S. (1993). Applying Problem Solving and Learning to Diagnosis. In P. S. Rosenbloom, J. E. Laird & A. Newell (Eds.), *The Soar Papers: Research on Integrated Intelligence* (pp. 674-687). Cambridge, MA: MIT Press.

Watkins, M. J. & Gardiner, J. M. (1979). An appreciation of generate-recognize theory of recall. *Journal of Verbal Learning and Verbal Behavior*, 18, 687-704.

Wray, R. E., Laird, J. E., Nuxoll, A., Stokes, D., Kerfoot, A. (2005). Synthetic adversaries for urban combat training. *AI Magazine*, 26, 82-92.

Yost, G. R. & Newell, A. (1988). Learning new tasks in Soar. In P. S. Rosenbloom, J. E. Laird & A. Newell (Eds.), *The Soar Papers: Research on Integrated Intelligence* (pp. 688-702). Cambridge, MA: MIT Press.

Zipf, G. K. (1949). *Human Behaviour and the Principle of Least Effort*. Cambridge, MA: Addison-Wesley.