Ritter, F. E. (1995). Review of "Soar: An architecture in perspective". *Philosophical Psychology*, *8*(3), 301-305.

Book review for *Philosophical Psychology*

Frank E. Ritter
Dept. of Psychology
U. of Nottingham
Nottingham NG7 2RD
ritter@psyc.nott.ac.uk
++ 44 (602) 515 292 (515 324 fax)

Soar: A cognitive architecture in perspective[1]

Michon, J. A., and Akyürek, A. (eds.)
Dordrecht (Germany), Boston, and London: Kluwer Academic Publishers.
xi+248 pages
1992.
ISBN 0-7923-1660-6
87

Many theories on psychology can be characterised as a string of boxes labelled "processing", "more processing", and "yet more processing." The idea that theories of human behaviour should predict that humans are intelligent and explain how this is possible by specifying the information processing to a level that can be performed automatically (and thus create intelligent agents) has not been as widely accepted in Europe as it has been in America. It is therefore a pleasure to note how a research group in Holland has embraced this notion, at least a particular version of it, and have extended it in ways that its American originators either did not anticipate or did not have the resources to explore. This is not surprising, for this prediction of intelligence has many implications and exploration in this area can be difficult.

This call for new theories in psychology that model the process of intelligence was started around 1960 with the Logic Theorist (Newell, Shaw, & Simon, 1958), a process model of logical proof creation. The model was implemented first as a set of logic rules summarising behaviour, but it was quickly modified so that a computer program could create the behaviour automatically. It defined a set of sufficient (but not necessarily necessary) knowledge that could be used to derive logic proofs. It was one of, if not the first theory that by processing information itself (although in a limited way) predicted that humans were intelligent, or at least as intelligent as the theory. Since then, this approach has been used to explain a variety of phenomena, ranging from solving simple puzzles to behaviour in medical diagnosis. The resulting models can be used to understand behaviour, but they also have many applied uses. For example, they have been used to create deployable models of expertise (expert systems), to create intelligent computer based tutoring systems, and to design curricula.

Cognitive architectures. The areas of coverage of a model need not be limited: Cognitive models offer a potential way of unifying and integrating all the regularities gathered by psychology, attempting to cover data from all areas of behaviour (not that we an do *that* yet) with a single set of mechanisms. This has taken the form of a call for *unified theories of cognition* (UTC). Work towards UTCs now takes the form of understanding what is common between tasks, the *cognitive architecture*, and what must be known to perform each individual task, task knowledge (Newell, 1990).

This represents a different approach to developing psychological theories. Instead of theories covering more and more about less and less, architectures for cognition must provide broad coverage

[1]I need to thank Todd Johnson, Josef Nerb and David Wood for useful comments.

of the data, initially covering more and more areas and behaviours to make sure that they have things mostly right and that the architecture can provide a complete basis for intelligence, and then only later going back to make sure that everything fits just so. (In reality, development is much more mixed, but things don't always, and shouldn't always change to cover a small data point if a large area must then be given up.) This approach does not deny the absolute importance of getting the details right, it just represents a difference in priorities, of what to approach first and the primacy of including the prediction of intelligence.

Multiple attempts at creating such architectures are necessary, and the two largest architectures for attempting this (Soar and Act-R) are each on their fifth or sixth major revision. Since this book has come out, the Soar architecture has seen one major overhaul. It now runs near real time (i.e., it takes about a second to model a second of behaviour), and several new interfaces are being developed to make model creation and interpretation easier. So some of the work presented in this book now appears to be dated (i.e., how not to end up learning to do everything at once), but most of what is reported is still true and will likely apply to other architectures.

It will take multiple groups and multiple architectures to understand what to do with such beasts, and nowhere is there a call for a single architecture, and there won't be for some time. This view of architecture and knowledge, offers a way to organise our theories, and gives us heart in knowing that the reasons things seem hard is that they are, for the material that psychologists study is in many ways harder and broader than that of any other science.

<u>Overview of the chapters.</u> This book provides a summary of work done with the Soar architecture (Rosenbloom, Laird, Newell, & McCarl, 1991) primarily at the University of Groningen in the Netherlands. The first chapter includes a brief (41 page) overview of Soar by Newell (whose honorary degree from Groningen prompted this book). Newell (1990) is more complete of course, but this chapter should be enough to gain a preliminary understanding of cognitive architectures with Soar as an exemplar, and makes the book relatively self-contained. Because of this chapter's relative clarity and simplicity, I will use it in the future in my classes. This alone makes the book worth reading because Newell clearly and directly addresses the contributions of Soar to psychology and provides an approach to measuring them.

Following this are three chapters on planning; a review chapter; a chapter laying out the details of a specific way to use a general planning method (means-ends analysis) using the features of the Soar architecture; and a report on a planning system that does multiple tasks at the same time, that is, driving. The final two chapters in the book each explore a new way to represent problem solving in the Soar architecture. The questions and preliminary answers addressed in these chapters will apply to many planning systems, not just those implemented in Soar. All of these chapters include the motivations behind building cognitive architectures as well as the specifics of a particular architecture.

In addition to the main chapters, there is also an introduction to the book and a introductory biography of Newell. These are useful in their own right because they provide interesting and grounding context to the history of Newell and Soar, and their relationship to the group in the Netherlands. I agree with Michon the author of the introductory biography of Newell, that the root of cognitive architectures can be traced back at least to Newell's early commentary on how to approach psychology (Newell, 1973), and incorporates multiple threads of his work. Later, autobiographical material corroborates their view.

<u>Newell's short introduction to Soar</u>

As part of his initial chapter, as an explanation of Soar, Newell lays out a theory of what having a UTC would look like within a simple experimental paradigm. This model would include a natural language parser to accept simple tasks and a planning system to create behaviour for testing immediate responses (such as judging that the star is above the box). This broad range of data coverage, from input to output, has not been realised often enough with Soar, but represents the heart of what unified theories are all about.

He also explains why enterprises like the development of cognitive architectures need large communities. Compared to other aspects of psychology, the machinery is relatively expensive to produce, and testing requires more resources than are typically available within a single investigator's research group. Although work continues on getting the architecture right, this has been restrained by interface issues (how to understand and use the current architectures) and the sheer size of the effort that these problems seductively hide and the difficulty of getting early attempts right. (Based on Michon's description on p. 15, one can draw an interesting analogy to IPL-V, an early high-level programming language. Even Newell, its author, needed a command reference card to use it.)

<u>The Nov scale.</u> Oddly, I think that the largest contribution of this chapter is the description language Newell provides to explain the unique contributions of Soar. To do this, he introduces the Nov scale (from *novus*, Latin for new), to assign ratings of importance to pieces of scientific work similar in meaning to the seismic Richter scale. The rating of a result is based on how widely it is felt and how much it shakes the field. The idea of assigning ratings to clarify the importance of discoveries and theories will help clarify our thinking about how science is done and about how results are aggregated and explained by theories, including process models. He explains, through examples using the scale, how to cut several Gordian knots that have dogged modelers who let themselves be entangled, such as how can you get more out of a simulation than just what you put in? Perhaps it should become the Newell scale.

I think he often over or understates many of the ratings, but there are at least two ratings of results of Soar that I particularly agree with: The methods within Soar that have evolved by necessity for deliberate learning of declarative facts provide an explanation why human declarative memory is necessarily slow to write and fast to read. This is a *5* on the Nov scale, a *first*, which will "be felt by all". The other result arises from the chunking learning mechanism being closely tied to problem solving. This leads to the learning rate being proportional to the rate of impasse creation, which is caused by lack of knowledge. This is a *6, a new result not empirically known.* I'm sympathetic to this, it has always seemed true to me that "you either do well or you learn a lot."

<u>Three related chapters on planning</u>

The first chapter on planning by Akyürek lays out a theoretical view of how to base the concepts of plans and planning more firmly in terms of an architecture with a strong theory of memory and ubiquitous learning that creates plans, uses them in an active way to solve problems, and can later retrieve them to assist in later planning. In a simplistic sense this is case-based reasoning, but because the theory incorporates the additional items noted here, it should be taken seriously indeed.

The second chapter, also by Akyürek, includes an example application that may be informative, and grounds what Newell meant by using an architecture. Akyürek presents a previously unseen simple and elegant mechanism for using architectural features to do means-ends planning that includes within-trial learning and transfer while planning. While not completely well commented, an illustrative program included as an appendix to this chapter makes it clearer what a Soar model looks like and how one works.

The third chapter on planning, by Aasman and Michon, describes a model that drives a simulated car. This chapter makes two important points. First, it illustrates the types of goals that can arise in the Soar architecture, and how to service them. This provides a grounded explanation of the difference between dynamic, lack-of-knowledge caused processing goals (how to do something), and more homeostatic goals, where an environmental value is being maintained. They do a good job explaining this given that there is a Soar structure also called a goal. Second, the chapter presents a theory of planning and problem solving that does multiple tasks and that is interruptible. This model shows how multiple information sources can be integrated to explain psychological phenomena. These are the facilities that the architecture was designed to support, but which has been little used in the New World until recently (e.g., Rosenbloom, et al., 1994).

<u>The final two chapters</u>

Aasman & Akyürek in the penultimate chapter explore a promising way to do problem solving in a more cognitively plausible manner. Many models in Soar, and one suspects in other architectures as well, choose between alternative actions through search, creating a hierarchy of potential actions, where later actions appear deeper in the hierarchy. In extreme cases, a large memory stack is created as the search space is sampled. This approach is not cognitively plausible.

They propose three alternatives that flatten this hierarchy by using existing mechanisms in new ways to reduce the hierarchy. Solutions 1 and 2 eliminate the use of a separate level that represents and remembers the tied operators to be evaluated by using smaller structures to refer to the results of operations. They do not completely remove the goal hierarchy, but make it smaller. I expect one of these approaches to be used in new models, for they both seem to dominate older approaches in terms of speed, time, and cognitive plausibility. Solution 3 is more complicated and more promising. It uses only one state and only considers a single path at a time, using explicit backtracking operators when necessary. Explicit and deliberate learning must also be performed to reject the bad operators. All of these changes make the problem solving more cognitively plausible by making learning more deliberate, the active memory smaller, and the amount of knowledge stored in long-term memory larger. All of these approaches offer an additional benefit, they are more interruptible; that is, when interrupted, they can get back to where they were faster than the current default approach to search. The only drawback to this chapter is that it necessarily uses some advanced concepts without explaining them in detail.

In the last chapter, Rosenbloom and Aasman examine how chunking can be used for explanation-based learning (EBL), a machine learning approach that integrates inputs from the world into a mental model using a domain theory to 'explain' the new data. They present a description of how deductive and inductive knowledge level learning (Newell, 1982) can be supported by the chunking mechanism in Soar, which is typically (but wrongly) viewed as simply a caching mechanism. Given the right knowledge, which leads to inductive procedures, the seemingly deductive chunking mechanism can spit out inductive rules. There are some unseemly or at least unexpected machinations and actions the architecture has to take to do EBL. For example, the chunking mechanism is automatic and represents what it learns as production rules, which are not available for inspection by the running model (they are, of course, inspectable by the analyst). Therefore, in order to do EBL, the algorithm must represent what is learned as an explicit declarative structure. The most interesting implication of this that they draw is that EBL is likely to be a reflective based type of learning behaviour, and not at all automatic. This also suggests that studying reflection may be important for understanding implicit/explicit learning. Their footnote, noting that all the behaviour examples have been implemented and are known to run, is laudatory.

Mixed in here by implication, is that new ways of using the chunking learning mechanism are still being found. This type of result, new uses of old mechanisms, is presented several times in this book. It tells us much about architectures, how they are used and how they are developed. Many of the mechanisms in Soar have many uses, and some of these undoubtedly remain to be discovered. As Newell (1990) once noted**,** there is more in your architecture than one would imagine. Any architecture, one may even conjecture any good architecture, will have this problem as well.

Laments

Where to improve Soar (although not necessarily how) is clear (this is less true for "box models" and mental models of psychologists), so the future looks good. We know enough now to indicate that this enterprise is worth taking seriously, and we know where we need to improve it and the models created in it (Grant, 1962). But there are also problems where the way forward is unclear and are usually long-standing enough to be labelled laments. The laments related to this book can be broken down into problems with Soar the theory and with its presentation.

Laments about Soar. The work presented here, like much work to date with cognitive architectures in general and with Soar as well, does not take advantage of (i.e., use) the complete architecture. 'Putting it all together', doing multiple tasks from input through cognition to realistic output is not presented here. When data is used to constrain model development or are compared with a model's performance, the number of regularities (out of Newell's estimate of 3000**)**, is much lower than one would want, on the order of 10 to 20 regularities. The game that Newell advocated, of having this fit recorded and bettered by later models ('Anything you can do I can do better') has not become a routine aspect of the work, presented here let alone of psychology. The interruptibility of the architecture (except for this chapter on driving) and the use of the sequential predictions of the models have been little used. And finally, the deployment of these models of expertise in applications, a possible reason to build such models, has not occurred as often as one might hope.

There are a few exceptions to these laments if one looks outside this book, but the generalisations stand. Perhaps these problems are simply the result of the size of the undertaking to bring everything together, but because they represent the direction and results of unified theories of cognition, continued attention must be paid to these laments and to how well they are being met.

Laments about the presentation here. Only a single lament about the presentation here is worth noting. All of the authors, starting on page 1, get "what Soar will do" wrong in spots. (As do many writers about many architectures.) Soar, in that sentence has several meanings, only one of which is "Soar the theory". When speaking at a Soar workshop, this phrase is acceptable because the other meanings are clear to the listeners, but when addressing a wider audience it leads to misunderstandings at best, and charges of hubris and chicanery at worst. The writer of such a phrase usually means "What Soar will do after a competent analyst has included the necessary task knowledge" or "what we as a community can accomplish in 2 months to 5 years time after modifying the basic architecture." Soar the program, the theory, or the community developing it, given a new task (e.g., writing music), will still initially do nothing. Without knowledge about the task, the architecture (which many people will read as the meaning of "Soar" in this case) will just sit there. (This is not to say that it should just sit there, but currently it does, although work goes on to get default behaviour to occur in such circumstances.) Creating Soar models remains difficult under optimal conditions and it needs to be as routine and straightforward as ANOVAS, which were not widely used until they could be done automatically without the analyst's hand or mind engaged. Writers about Soar should be clearer when using this phrase, separating what is already known or done from conjecture and hopes for their architecture.

Summary

This book presents Soar with a European style, with some attention to detail and scholarship (e.g., an author and subject index); it includes more references than is typical, and each chapter clearly situates itself with respect to other work. There are only two flaws (a) the awkward placement of footnotes at the end of chapters is inexcusable; and (b) the U. of Groningen group has broken up, alas. While they carry on their work at other research centres in the Netherlands, they are not as focused.

An annotated list of further readings and resources appears at the end of the book, serving as a natural conclusion. This list omits the two shortest and most approachable references (Waldrop, 1988a, 1988b). Since this book's publication, two further resources have come available, the Brain and Behavioral Sciences target article and commentary (1992) and a two volume set of edited papers on the Soar project (Rosenbloom, Laird, & Newell, 1992). They are both very complete, but I suspect they may not serve as well as a gentle introduction.

So who should read this book? I can see three groups (a) philosophers who want a simple and complete introduction to what process models provide psychology and theories of mind; (b) psychologists who are searching for a more unified way to approach their field, and would like to have theories that predict that we are intelligent; and (c) Soar researchers who I suspect have not read and used this book as much as it deserves.

Psychological regularities continue to grow in Newell's absence (he died in 1992), at least at 90 per year (e.g., 30 journals x 30 papers per year by 10 papers to establish a regularity). The need to put them altogether is only becoming more important.

FRANK E. RITTER
RITTER@PSYC.NOTT.AC.UK
UNIVERSITY OF NOTTINGHAM

References

Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models. *Psychological Review, 69*, 54-61.

Newell, A. (1973). You can't play 20 questions with nature and win. In W. G. Chase (Ed.), *Visual information processing* (pp. 283-308). NY, NY: Academic Press.
Newell, A. (1982). The knowledge level. *Artificial Intelligence, 18*, 87-127.
Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
Newell, A. (1991). *Desires and diversions: Carnegie-Mellon University School of Computer Science Distinguished Lecture* (64 min. video tape). Palo Alto, CA: University Communications.
Newell, A. (1992). Précis of *Unified theories of cognition*. *Behavioral and Brain Sciences, 15*, 425-492.
Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review, 65*, 151-166.
Rosenbloom, P. S., Johnson, W. L., Jones, R. M., Koss, F., Laird, J. E., Lehman, J. F., Rubinoff, R., Schwamb, K. B., & Tambe, M. (1994). Intelligent automated agents for tactical air simulation: A progress report. In *Proceedings of the Fourth Conference on computer generated forces and behavioral representation*, . Gainsville: U. of Central Florida.
Rosenbloom, P. S., Laird, J. E., & Newell, A. (1992). *The Soar papers: Research on integrated intelligence*. Cambridge, MA: MIT Press.
Rosenbloom, P. S., Laird, J. E., Newell, A., & McCarl, R. (1991). A preliminary analysis of the Soar architecture as a basis for general intelligence. *Artificial Intelligence, 47*(1-3), 289-325.
Waldrop, M. M. (1988a). Toward a unified theory of cognition. *Science, 241*, 27-29.
Waldrop, M. M. (1988b). Soar: A unified theory of cognition? *Science, 241*, 296-298.