AI Magazine, 23(3[Fall]), 107-109.

A review of the 22nd Soar Workshop

Frank E. Ritter and Isaac G. Councill

School of Information Sciences and Technology
Penn State
University Park, PA 16801
ritter@ist.psu.edu

Introduction

Soar is one of the oldest and largest AI development efforts, starting formally in 1983. It has been proposed as a unified theory of cognition (Newell, 1990). Most of its current development is as an AI programming language, and this was evident at the 22nd Soar Workshop held at Soar Technology near the University of Michigan in Ann Arbor, Michigan on 1 and 2 June 2002.

The work presented at the Soar workshop illustrated several trends that should be of interest to those outside of the Soar community, and some of the issues that Soar is grappling with that other groups developing unified theories of cognition or AI programming languages may find interesting as well.

Soar workshops are currently held annually in late May or early June. Recently they have been held at the University of Michigan and at the University of Southern California's Information Sciences Institute, two of the largest Soar sites. They are held on a Saturday and Sunday, with a tutorial or two on the preceding Friday.

This year the workshop was preceded by two days of tutorials: an introductory tutorial on Thursday and a more advanced tutorial on Friday. The tutorials were held at the University of Michigan's Advanced Technology Lab, and at the workshop site. The workshop was held just off-campus from the university at Soar Technology, a four-year-old start up company that produces Soar systems for industry and government. This is a good sign, we think, for it indicates a certain maturity of Soar. This site also indicates that the largest Soar site, as least as indicated by speakers at the workshop, is now a research and development company.

The Soar workshop assumes that the audience is already familiar with Soar (thus the tutorials). The talks are short, sometimes as short at 5 minutes, and rarely longer than 15 minutes. The proceedings typically are just copies of the slides used for the presentations, but sometimes more extensive prose or examples are included. In the last several years these have been put up on the web, Soar Workshop 21 (2001): ai.eecs.umich.edu/soar/workshop21/talks/; Soar Workshop 20 (2000): www.isi.edu/soar/soar-workshop/proceedings.html. This year's proceedings will appear shortly at www.soartech.com.

There were 37 talks this year as well as a discussion session and 57 attendees. The speakers came from 11 sites (including 2 from Britain), but the talks were not evenly distributed as 13 came from Soar Technology, 10 from Michigan, 4 from USC/ISI, and 2 from Penn State. 7 sites made one presentation, sometimes representing multiple researchers. The community is thus primarily at the three large sites, but with numerous small sites.

Highlights of the workshop

There were numerous interesting pieces of work that we have organised into ten themes. This is our take on the workshop and it reflects our biases and interests.

The proceedings, based on a short discussion at the end of a session, will continue to include just slides, not full papers. This provides a very good value yearly summary, but with some obvious defects. The format provides a lightweight way to describe ongoing work as well as projects that are about to start, which encourages collaboration.

Modifying the architecture

One talk that stood out was a talk about adding working memory decay to Soar (James). A Soar model of the Tower of Hanoi was run in a version of Soar modified to include decay of memory objects. The resulting behaviour was compared to existing data. The behaviour of the model in the modified architecture correlated 0.986. The behaviour of a comparable ACT-R model of the Tower of Hanoi correlated 0.995, virtually identically well.

Another interesting project was starting to examine how to include a range of behavioural moderators into the Soar architecture by adding a connectionist substrate (Jones). This substrate will include measures of pain/pleasure, arousal, and confusion/clarity of the situation.

Some work was introduced exploring forgetting, examining what happens to response time if learned rules that are not used are discarded after various amounts of time (Kennedy). The system appeared to get slower as more was learned, but sped up if very infrequently learned rules were deleted. It did not speed up if unused rules were deleted just a very short time after being learned. Removal of learned rules has not been optimised in Soar, making this result hard to interpret. What was interesting is that the results to us appear to be consistent with Doorenbos's results, that there is little slowdown as new rules are learned (Doorenbos, Tambe, & Newell, 1992).

Learning

There were several projects to include new approaches or types of learning. These projects, some in an early stage, were to include and integrate learning by observation using inductive logic programming (Konik; Hawkins), learning through self-explanation (Jones), reusing and testing the previous symbolic concept acquisition model (Wray), making learning and behaviour more robust (Kiessel; Nielsen; Beard), learning how to recover from lack of knowledge but with some domain specific yet slightly general knowledge (Kiessel), and a project attempting to discover errors in behaviour by having agents catch their own errors (Wallace).

Multi-agent systems

Many of the projects that were about modeling human or agent behaviour also concerned themselves with multiple agents. This included models of teamwork, communication between agents, and how to motivate other agents (respectively: Nuxoll; Beisaw; Taylor). Another interesting project examined conflict resolution and the possible dilemmas that can arise when problem solving is not rational (Kalus).

Computer generated forces

The largest current (but certainly not only) application of Soar is for computer generated forces, that is, modeling humans and combat vehicles in military simulations. This was evident in the large number of talks at the workshop in this area. In addition to being the application domain of several other talks, work was presented on modeling urban combat with Soar in OneSAF (a simulation tool for synthetic environments), and in a variety of other projects (Reece; Wray; van Lent; Jones; Beisaw; Wallace).

Games

Laird believes that computer games are an important and useful application for AI models (Laird & van Lent, 2001). Several folks working with Laird presented work on creating Soar models for populating computer games, including a haunted house game (Laird) and military tactics games, both as an individual in a first-person shooter like Unreal Tournament as well as small squad games (van Lent). Agents are being built to serve as opponents and sometimes assistants and colleagues. Exploratory work is being conducted that aims to put Soar into the role of an interactive story director (Magerko) as well as providing directable characters within the story (Assanie). Some work was presented dealing with interfacing Soar to the Unreal game environment (Stokes and Kerfoot).

Interfaces for programs

Several talks reported work to provide various types and levels of application program interfaces to Soar. These APIs are to support projects that need to embed Soar and to provide a more uniform and steady approach to building tools that work with and for Soar. None have been released, but two are promised shortly (Wessling; Pearson).

Interfaces for modelers and users

Many in the Soar community are realising that interfaces for teaching users and developing Soar models (broadly defined) need to be improved. In addition to discussions in the hall, in the discussion session, and at two lovely evening socials, several of the papers were about user interfaces or had aspects related to user interfaces. Several groups are working on extending programmer interfaces to Soar (Harleton; Crossman). Interface usability concerns were addressed with a task analysis (Ritter), a new toolkit for creating displays for Soar models was presented (Taylor), and at least two groups are working on adding explanation capabilities to Soar (van Lent; Councill).

Reuse

One of the main reasons for working within a cognitive architecture, according to Newell, was to encourage theory cumulation and reuse. In a cognitive architecture like Soar, as theories are represented as programs, either as production rules (knowledge), or as changes to the architecture (primitive capabilities) this reuse has to mean at least the reuse of ideas, but more likely needs to mean the reuse of systems and knowledge sets. We believe that reuse has been a continual problem in Soar.

There are examples of reuse within the Soar community, and a few were presented here. A theory of concept acquisition was reused and is being extended to match a large pot of data for a cognitive model comparison exercise (Wray).

NL-Soar, a theory of how natural language is realised in Soar, continues to be developed (Lonsdale). When it is complete, and available, it offers to be a widely used (and thus reused) theory of natural language understanding.

More than one of the speakers reported a new and clearly useful tool for writing/editing/augmenting Soar code. For the last speaker to do this, an audience member, one of the authors, asked the speaker what the take-home message was. Was it that the tool would be released, notes on the technique would be made available, or was he just teasing us? For far too many reports (including one of ours), so far it appears that the answer was the last one.

Models of eyes and hands

A model was reported that interacted with an interface based on the picture on the screen (van Lent). Image processing was used to derive environmental structure, that is, where buildings were and where doorways were in a synthetic environment. This spatial map was then used in several ways to generate motion through the environment.

Interacting based on working through the display is a useful step for making Soar less dependent on instrumented ties to particular interfaces. This approach appears to be similar to St. Amants' SegMan approach (St. Amant & Riedl, 2001), but the approach discussed by van Lent was still tied lightly to that synthetic environment.

Robotics

A Soar model is being prepared to fly an unmanned air vehicle in the UK (Smith). The plane has been selected, and the initial Soar model is currently generating and flying routes in a simulation. This is exciting as it shows another autonomous application of Soar. Also reported with this project was a clever hack for quickly loading information into Soar by parsing a map database into Soar rules that augment Soar's initial working memory with the contents of the database as Soar starts up. Another talk introduced a small robot whose small head (about 32 k of RAM!) was filled with Soar, was able to follow a line on the speaker's table (Waterson).

The future

Work on Soar continues to present new applications and new algorithms for learning and behaviour in a variety of areas. In this sense, work with Soar is progressing and rightfully has the attention of people who do not directly use Soar.

Cognitive models, in addition to the AI projects, are still occurring within Soar. There were models of concept acquisition, natural language recognition, and problem solving. This work has not disappeared; it just appears in slightly different form.

One of the papers compared Soar with respect to BDI architectures and to GOMS, noting many commonalities of component structure, particularly how these perspectives share similar architectural components on the problem space level (Jones). The analysis highlighted several aspects of behaviour that may be currently missing from many of these approaches, such as parallel active goals.

At the same time, there are areas where Newell's dream is not being realised, that are difficult for all groups working on cognitive architecture, where the answers are not easy. Theory cumulation, model reuse, and interfaces that promote the spread and uptake of this scientific approach are important ones we believe (we also work in the area of HCI, so we are particularly interested in these issues). Newell said, and we agree, that science like politics is the art of the possible. Soar does not have enough positive examples in these areas. The few good things in Soar that other communities can emulate include a variety of mailing lists (soar-requests@umich.edu for inquiries and subscriptions) and the help that often arises from diverse members of the community from requests to the lists, the annual workshop, central source code maintenance, a web site (ai.eecs.umich.edu/soar/), and a frequently asked questions list (acs.ist.psu.edu/soar-faq/).

If you would like to know more, the proceedings will be online at the Soar web site. If you would like to learn more in a hands-on format, a tutorial will be offered at Cognitive Science in August 2002 and perhaps at the International Conference on Cognitive Modeling in Bamberg, Germany in April 2003 (iccm2003.ppp.uni-bamberg.de). More tutorials and additional types of sessions will be offered at the next Soar workshop, to be held at the University of Michigan in 2003. As this will be Soar's 20th anniversary, extended tutorials and commentaries are being planned -- a festival of Soar perhaps.

Acknowledgements

Support for this report was provided by the US Office of Navy Research, award number N00014-02-1-0021. The views expressed in this article do not necessarily reflect the

positions or the policies of the U.S. Government, and no official endorsement should be inferred.

References

Doorenbos, R., Tambe, M., & Newell, A. (1992). Learning 10,000 Chunks: What's it like out there? In *Tenth National Conference on Artificial Intelligence*. 830-836. Menlo Park, CA: American Association for Artificial Intelligence.

Laird, J., & van Lent, M. (2001). Interactive Computer Games: Human-level AI's Killer Application, ai.eecs.umich.edu/people/laird/papers/AI-games.pdf.

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

St. Amant, R., & Riedl, M. O. (2001). A perception/action substrate for cognitive modeling in HCI. *International Journal of Human-Computer Studies, 55*, 15-39.