

Soar 8.3 Release Notes

- Fatigue/Timing Model from Randy Jones added
- Generic capture/replay facility added (beta)
 - permits exact replay of Soar Input from external simulation
- O-support mode 3: operator augmentation gets I-support
- TSI (Tcl-Soar Interface) upgrades and new options (v3.1)
- Sample RHS function definitions
- New demos for learning, planning, and I/O methods
- Several bug fixes:
 - timers (274) and stats cmd (240)
 - extraneous log file msgs (248)
 - Makefile dependencies (276 & 277)
 - RHS function support (280)
 - compiler warnings (mostly void ptrs)
 - memory leaks

Fatigue / Timing Model

(Randy Jones)

- Can regulate “real-time” decision cycle timing by setting variable **real_time_per_decision** (value in milliseconds)
- 3 new interface commands to support “attention lapsing” by modifying Soar I/O processing:
 - **start-attention-lapse msec**: sets the "attention-lapsing" variable to TRUE (1), and starts tracking amount of real time that should pass before ending lapse. Duration of lapse is number of milliseconds specified by argument. Not normally be called by user or agent (attention lapses normally get started automatically by the “architecture”),
 - **wake-from-attention-lapse** primarily intended to be called from the RHS of a production rule. This sets "attention-lapsing" variable to FALSE (0), and starts tracking amount of real time that has passed since last lapse.
 - **attention-lapse (-on | -off)**: boolean for whether or not attentional lapsing will occur.
- Brief AppNote from Randy Jones available on web page.

Capture / Replay Input Facility

- Allows for recording of all Soar Input Cycle commands, including *add-wme* and *remove-wme* events for every execution cycle, so later can replay exact input sequence.
- Two new interface commands:
 - **capture-input** [-open *filename* | -query | -close] starts and stops the recording (to file) of input cycle commands
 - **replay-input** [-open *filename* | -query | -close] reads input cycle wme changes from file and registers callback to execute wme changes during specific decision cycle. Alerts user if discrepancies arise between recorded timetag and newly-generated timetag.
- Facilitates timing experiments and debugging with external simulations

O-support mode 3: Operator augmentation gets I-support

- New default Support calculation for Soar 8.3. Old modes (0,1,2) still available.
- Similar to mode 2 (Soar 8.2 default) **except** that productions which include operator augmentations now get I-Support, eventhough they test the operator.

Eg: this production will get I-support in Soar 8.3:

```
sp {get-i tem*elaborate*in-room*visible
    (state <s> ^i o. input-link. entry. item <i tem>
        ^operator <o>)
    (<i tem> ^in-current-room true)
    (<o> ^name get-i tem)
-->
    (<o> ^place in-room
        ^i tem <i tem>)
}
```

TSI (Tcl-Soar Interface) upgrades

Tabs for displaying info windows.
All data preserved during run.

The screenshot shows a window titled "red Agent" with a menu bar (File, Options) and a tabbed interface. The tabs include "TSI Console", "Working Memory Elements", "State Stack", "Production Match", "Chunks", and "Justifications". The "TSI Console" tab is active, displaying a log of file loading operations. Below the console is another menu bar (File, Show, Memory, Productions, Watch, View, Commands, Demos, Help). At the bottom, there are three main sections: "Environment" with "Step", "Run", and "Halt" buttons; "Agent Only" with "Phase" and "Full Cycle" buttons; and "Options" with a checked "Active Update" checkbox. To the right of these is a "Soar Phase indicator" showing a cycle of I, P, D, A, O around a "WORLD" box. The status bar at the very bottom shows "Last Phase: NONE", "Learn: off all-levels", "Soar v8.2", and "Debugger v1.0b".

Original TSI Agent Window and Menubar

Application Control Buttons

Ability to run this agent only.

Can disable active updating for better performance.

Soar Phase indicator

Status/version

TSI (Tcl-Soar Interface) upgrades cont.

- TSI Version 3.1 requires at least Tcl 8.0
- Can set variable to revert to previous TSI agent window appearance and capabilities.
- Much easier to examine states and debug code, but at some cost of performance.
- Gives novices a better picture of Soar execution.

New demos

- Learning
- Planning
- I/O methodology and example
- all are found under the TSI “demos” pulldown menu

Sample RHS Function Definitions

- *rhsfun_examples.c* included in interface folder and linked in distribution
- shows 4 new RHS functions and how to define them and add them to Soar.

Several bug fixes

- timers repaired(274)
- stats cmd includes correct timing info (240)
- extraneous log file msgs removed (248)
- Makefile dependencies corrected (276 & 277)
- RHS function management patched (280)
- compiler warnings addressed (mostly void ptrs)
- memory leaks patched