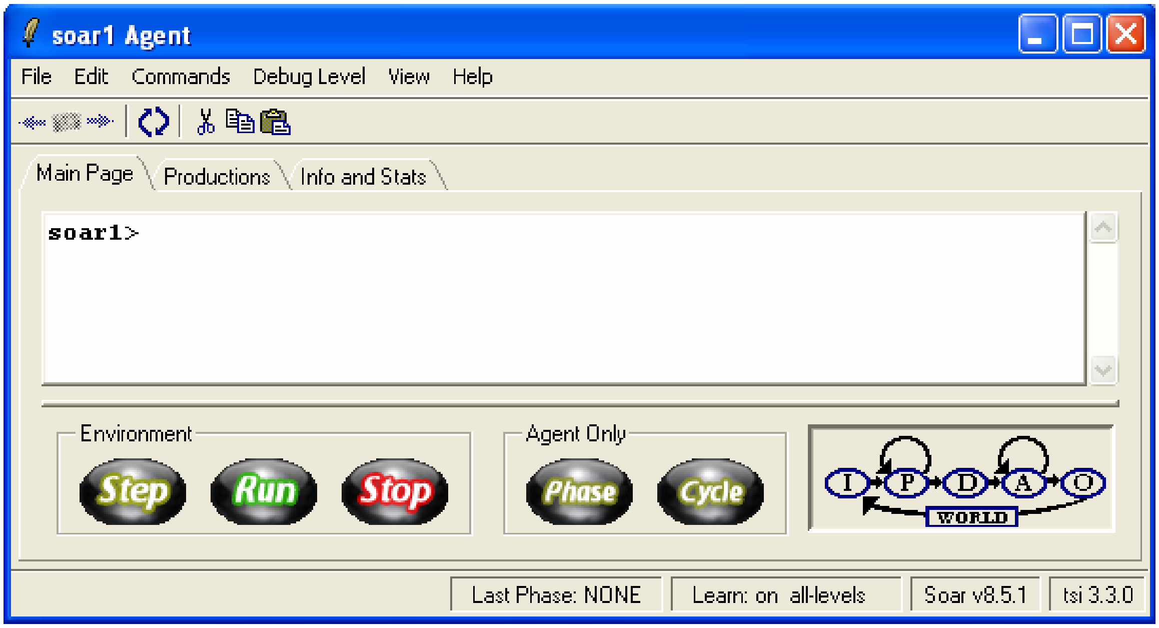
TSDebugger 4.0 Tour/Manual

Starting up the TSDebugger produces a window which looks like this:



```
soar1>
```

Last Phase: NONE | Learn: on all-levels | Soar v8.5.1 | tsi 3.3.0

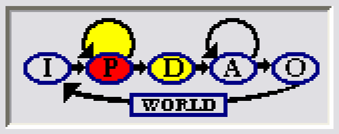
The title bar shows the agent's name. The bar at the bottom shows the last phase run in the decision cycle, the current learn setting (for Soar's chunking algorithm), the Soar kernel version and the TSDebugger version. The toolbar below the menus provide icons you can use to perform common editing commands such as cutting, copying and pasting as well as refresh the content of the current display, if need be.

The buttons along the bottom can be used to make the agent run or stop. If you're agent is connected to an environment, the three buttons on the left allow you to step, run and stop agent in lock-step with other agents and the environment. The two buttons on the right, allow you step the agent through either individual phases or decision cycles independent of other agents and the environment.

The graphic in the lower right shows the decision cycle, and it lights up to show where the agent is in the cycle.

- **I**: Input phase
- **P**: Proposal phase
- **D**: Decision phase
- **A**: Application phase
- **O**: Output phase
- **WORLD**: The point at which the environment, if it exists, will be updated.

The last phase that the agent is in will be highlighted in red, while the next phase the agent will execute will shine yellow. Some phases can repeat before proceeding to the next phase, namely proposal and application phases; when in those phases, you will notice that both the arrow and the subsequent phase will shine yellow.



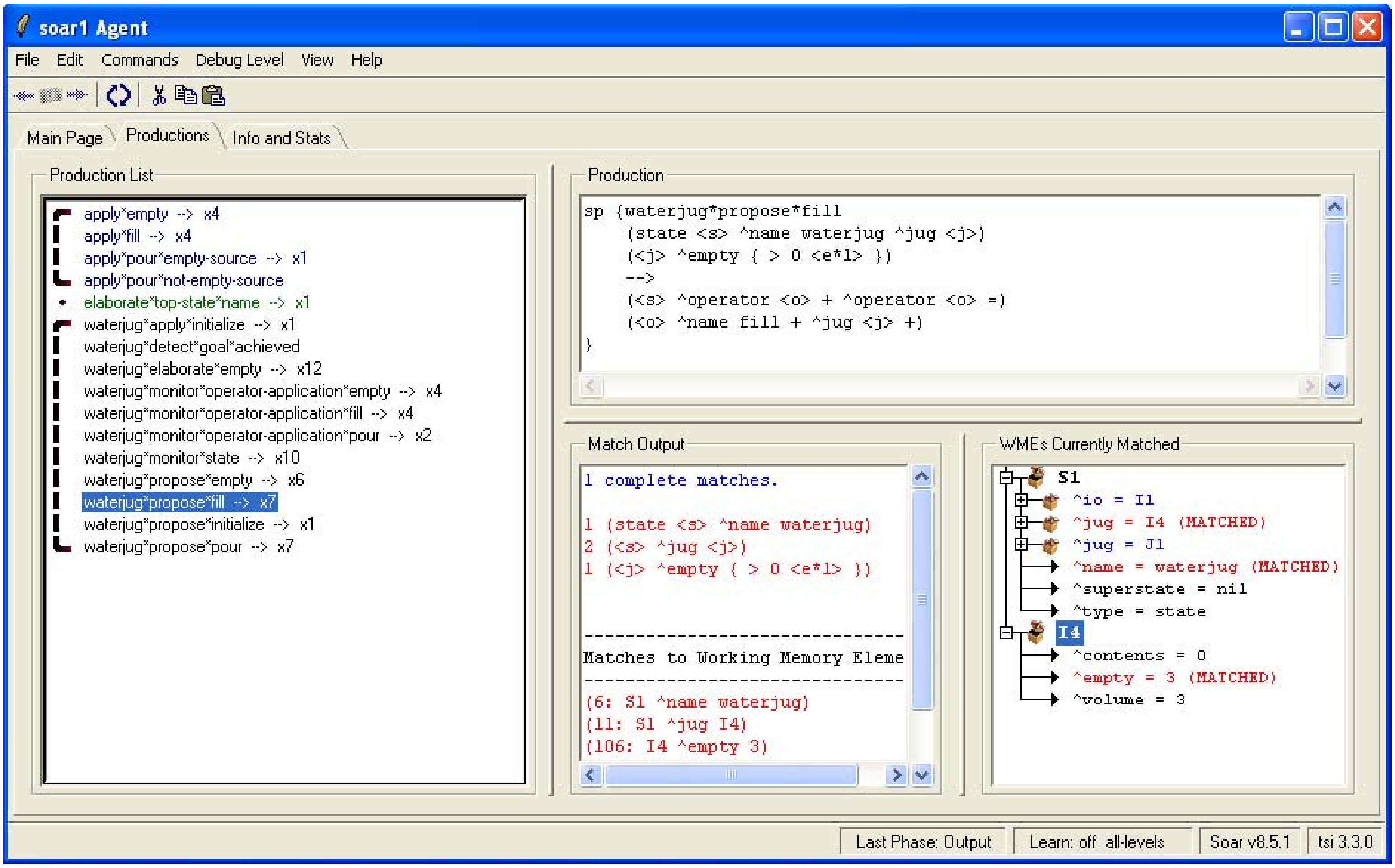
There are three tabs along the top that allow you to switch to different "notebook pages" that display various information about your agent. The three notebooks are :the Main page, the Production page, and Info and Stats page.

- The Main page always contains a command-line that lets you load and run agents, as well as execute any of the Soar commands listed in the Soar Manual. The Main page may also various other powerful, informational panes that you can use to help understand and debug your agents. These will be covered later.
- The Productions page displays information about the productions, or rules, that constitute your agent. You can view the contents of those rules as well as see how they correspond to the current state of the agent.
- The Info and Stats page compiles various data about both your agent and the current settings of Soar.

For the purposes of this tour, we will load the waterjug example from the demos directory and run it for several decision cycles. This will give us some information to look at on the other tabs.

Productions Tab

The Productions tab looks like this:



```
apply*empty --> x4
apply*fill --> x4
apply*pour*empty-source --> x1
apply*pour*not-empty-source
elaborate*top-state*name --> x1
waterjug*apply*initialize --> x1
waterjug*detect*goal*achieved
waterjug*elaborate*empty --> x12
waterjug*monitor*operator-application*empty --> x4
waterjug*monitor*operator-application*fill --> x4
waterjug*monitor*operator-application*pour --> x2
waterjug*monitor*state --> x10
waterjug*propose*empty --> x6
waterjug*propose*fill --> x7
waterjug*propose*initialize --> x1
waterjug*propose*pour --> x7
```

```
sp {waterjug*propose*fill
    (state <s> ^name waterjug ^jug <j>)
    (<j> ^empty { > 0 <e*l> })
    -->
    (<s> ^operator <o> + ^operator <o> =)
    (<o> ^name fill + ^jug <j> +)
}
```

```
1 complete matches.

1 (state <s> ^name waterjug)
2 (<s> ^jug <j>)
1 (<j> ^empty { > 0 <e*l> })


--------------------------------
Matches to Working Memory Eleme
--------------------------------
(6: S1 ^name waterjug)
(11: S1 ^jug I4)
(106: I4 ^empty 3)
```

```
S1
  ^io = I1
  ^jug = I4 (MATCHED)
  ^jug = J1
  ^name = waterjug (MATCHED)
  ^superstate = nil
  ^type = state
I4
  ^contents = 0
  ^empty = 3 (MATCHED)
  ^volume = 3
```

Note that this pane is intended to be used when the agent's execution is stopped. The information within it will be updated every time you switch to this page. The following sections will described the contents of the various panes.

The Production List Pane:

The left pane holds a list of all productions which are loaded, grouped by the word or phrase which comes before the first asterisk. The color coding and grouping brackets are just for readability. If you use common operator naming conventions, this should help you find what you are looking for more quickly.

Firing Counts: Next to each production name is a number, for e.g. x7 by waterjug*propose*pour, which indicates how many times that production has fired in the current execution of the agent. A production that has never fired will not have a number next to it. You can use the firing counts to get a quick picture of what the agent has done in the past.

Breakpoints: Double-click a production name to set a breakpoint on that production. You will see an exclamation point icon ⚠ next to the production name when you do this. Next time you run the agent and that production match, execution will stop, allowing you to examine the agent at your leisure.. You will see message in the Soar console window in the Main page indicating that a breakpoint has been reached.

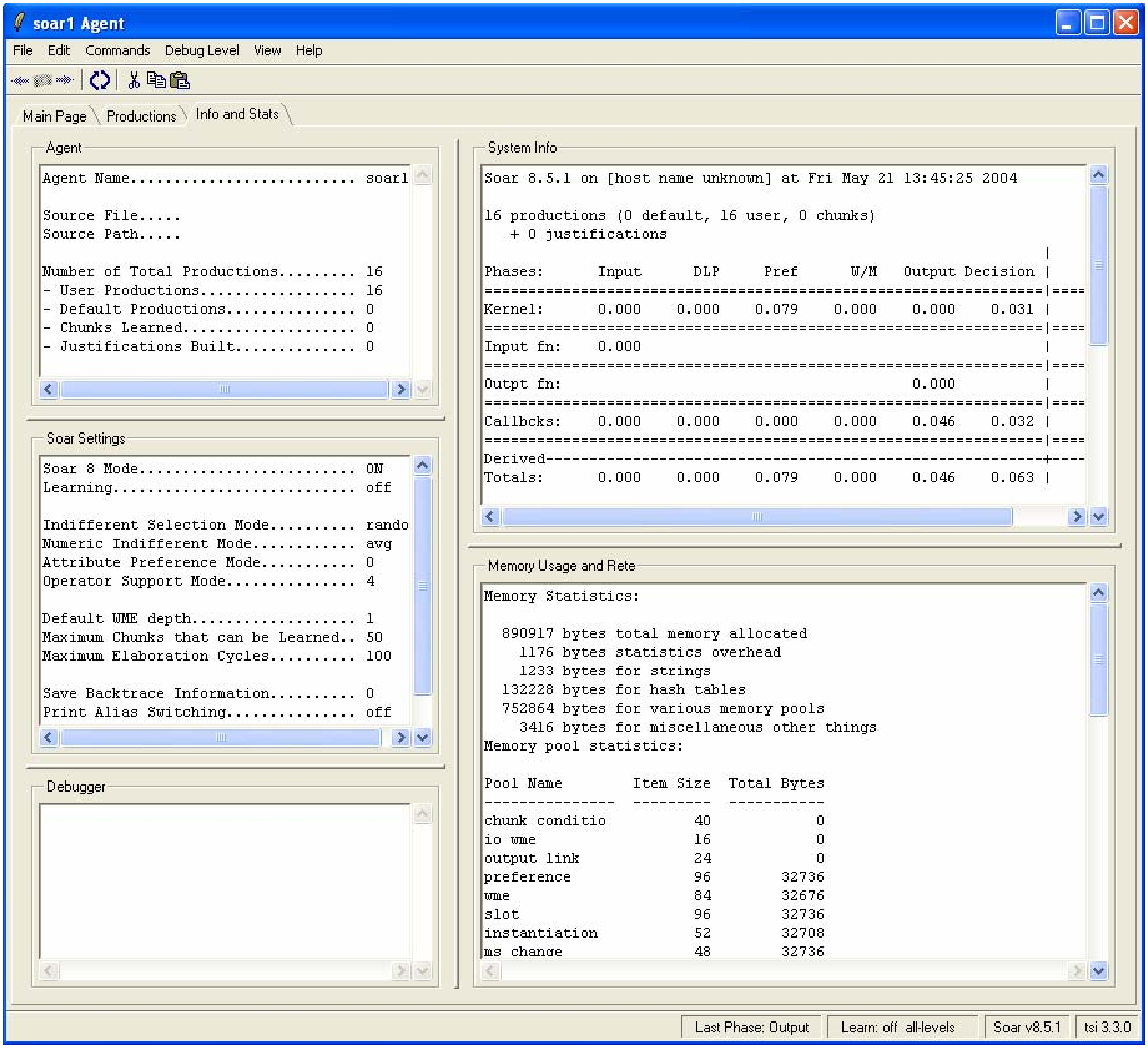
The Other Production Panes

Selecting a production will fill the other three panes with information about this production.

- The top pane shows the actual content of the production. Note that this display will not be identical to what you have in your source files. This output is the internal version that Soar uses. Variable names and certain condition structures may look different (though they are functionally identical.)
- The bottom-center pane shows the complete match status of the selected production with the agent's working memory. This is equivalent to the "matches –wmes" command.
- The bottom-right pane shows a list of all working memory elements in the current run that match the conditions of the chosen production. The actual attribute value pairs that match are highlighted in red and marked with the words (MATCHED.)

Info and Stats Tab

The Info and Stats tab looks like this:



Agent

```
Agent Name......................... soar1

Source File.....
Source Path.....

Number of Total Productions......... 16
- User Productions................. 16
- Default Productions.............. 0
- Chunks Learned................... 0
- Justifications Built............. 0
```

Soar Settings

```
Soar 8 Mode........................ ON
Learning........................... off

Indifferent Selection Mode......... rando
Numeric Indifferent Mode........... avg
Attribute Preference Mode.......... 0
Operator Support Mode.............. 4

Default WME depth.................. 1
Maximum Chunks that can be Learned.. 50
Maximum Elaboration Cycles......... 100

Save Backtrace Information......... 0
Print Alias Switching.............. off
```

Debugger

System Info

```
Soar 8.5.1 on [host name unknown] at Fri May 21 13:45:25 2004

16 productions (0 default, 16 user, 0 chunks)
   + 0 justifications
                                                                   |
Phases:      Input      DLP      Pref      W/M   Output Decision |
=================================================================|====
Kernel:      0.000     0.000     0.079     0.000     0.000     0.031 |
=================================================================|====
Input fn:    0.000                                                   |
=================================================================|====
Outpt fn:                                                0.000       |
=================================================================|====
Callbcks:    0.000     0.000     0.000     0.000     0.046     0.032 |
=================================================================|====
Derived----------------------------------------------------------+----
Totals:      0.000     0.000     0.079     0.000     0.046     0.063 |
```

Memory Usage and Rete

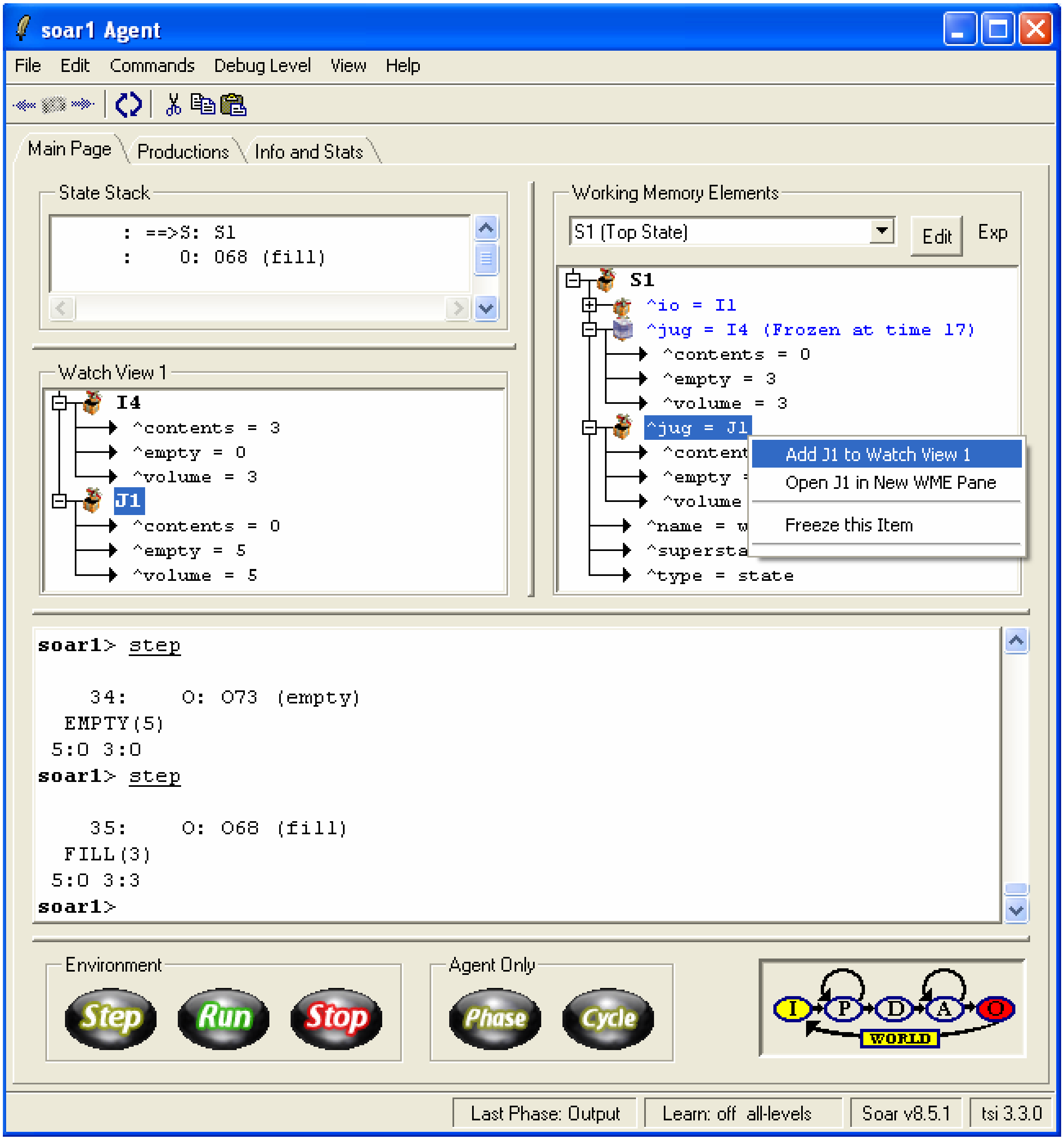
```
Memory Statistics:

  890917 bytes total memory allocated
    1176 bytes statistics overhead
    1233 bytes for strings
  132228 bytes for hash tables
  752864 bytes for various memory pools
    3416 bytes for miscellaneous other things
Memory pool statistics:

Pool Name        Item Size  Total Bytes
---------------  ---------  -----------
chunk conditio          40            0
io wme                  16            0
output link             24            0
preference              96        32736
wme                     84        32676
slot                    96        32736
instantiation           52        32708
ms change               48        32736
```

Some of the information is cut off because not all of it would fit in a readable screenshot.

- The upper-left pane shows some information about the current agent.
- The middle-left pane shows Soar's current settings.
- The upper-right pane shows various system statistics related to running Soar (there's actually more in that window than currently shows in the image).
- The lower-right pane shows other statistics related to memory usage and the Rete (again, not all of it is showing in this image).

Revisiting the Main Page

When we originally described the Main notebook page, we alluded to the fact that there can be various informational panes within that page. When the TSDebugger is first launched, it will use what is called the "minimal" layout. This layout consists of only one pane, namely the Soar console window. By going to the View menu, we can change to other layouts that contain more panes. The next, slightly more complex layout is called the Simple layout and looks like this:



```
      : ==>S: S1
      :    O: 068 (fill)
```

```
I4
  ^contents = 3
  ^empty = 0
  ^volume = 3
J1
  ^contents = 0
  ^empty = 5
  ^volume = 5
```

```
S1
  ^io = I1
  ^jug = I4 (Frozen at time 17)
    ^contents = 0
    ^empty = 3
    ^volume = 3
  ^jug = J1
    ^content
    ^empty =
    ^volume
  ^name = w
  ^supersta
  ^type = state
```

```
soar1> step

   34:    O: O73 (empty)
  EMPTY(5)
 5:0 3:0
soar1> step

   35:    O: O68 (fill)
  FILL(3)
 5:0 3:3
soar1>
```

The Soar console and execution panes are still at the bottom, but now there are three new panes. If you switch to the default, or "More Watch Panes" layouts, you will find even more types of panes. The follow sections will describe the various contents that can fill a pane and how they interact.

Pane Types

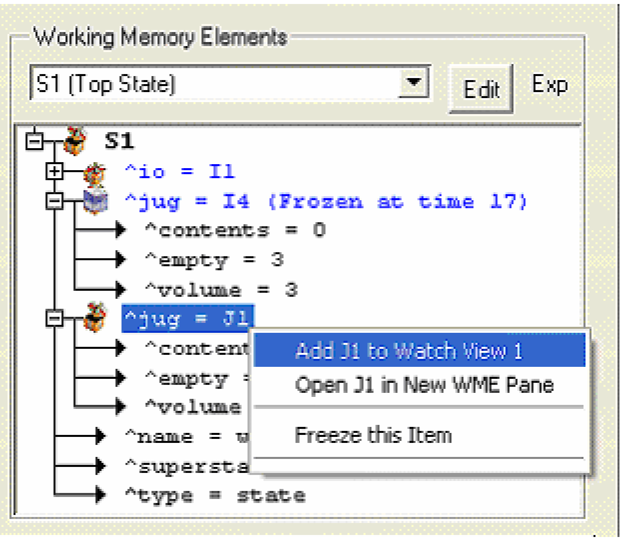
Currently there are seven different types of panes.

Basic Panes

Every layout will contain the following two panes

- The Soar console pane (1) – A traditional terminal in which you can type arbitrary Soar commands.
- Execution pane (2) – Contains the step/run/stop buttons and current phase diagram.

The Working Memory Pane (3)

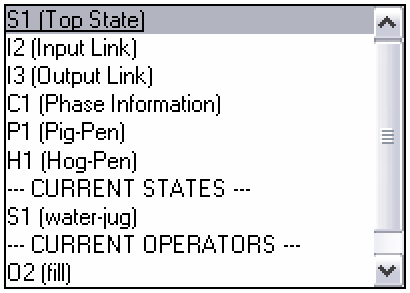


```
S1
  ^io = I1
  ^jug = I4 (Frozen at time 17)
    ^contents = 0
    ^empty = 3
    ^volume = 3
  ^jug = J1
    ^content
    ^empty =
    ^volume
  ^name = w
  ^supersta
  ^type = state
```

The top-right pane shows the contents of working memory in a tree form. Note that informational icons appear next to working memory elements (WMEs) that can give you information about their contents and any special characteristics you may have applied to their display. There are a lot of different things you can do in this plane.

Root Node Selection

The top portion of the pane allows you to specify the root WME of the tree displayed. The left-most item is an editable spinbox that probably has more functionality than you may initially assume. If you click on it or the triangle next to it, you will be presented with a list of possible working memory elements that you can make the new root of the tree. This list is dynamically generated every time you click on it. For example, here's one possible list:



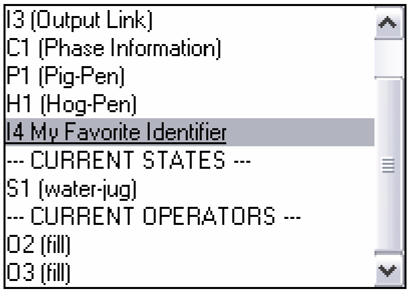
S1 (Top State)
I2 (Input Link)
I3 (Output Link)
C1 (Phase Information)
P1 (Pig-Pen)
H1 (Hog-Pen)
--- CURRENT STATES ---
S1 (water-jug)
--- CURRENT OPERATORS ---
O2 (fill)

If you examine this list closely, you'll see that there are three types of items.

1. The first section contains a list of **custom user-specified WMEs** to use as the root of the tree.
2. The second section contains a list of **all current states** within the agent. You cannot edit items on this part of the list. They are dynamically generated.
3. The third section contains a list of **all currently proposed or selected operators** in the agent. You also cannot edit items on this part of the list.

To select a new tree root WME, simply select it. To add a new item to the list, select the text of the currently selected item and start typing. When you hit return, the tsDebugger will parse what you typed. If the first word of what you type is a valid Soar WME, it will set the root of the tree to that WME. It will also add your new entry to the list of

custom user-specified WMEs in the menu we previously discussed. Any words after the initial identifier can be used as a mnemonic to remember what that item points to. For example, if you click on spinbox, select the text, type in "I4 My Favorite Soar Identifier" and hit return, the tree will be re-rooted on I4 and the menu will now look like this:



```
I3 (Output Link)
C1 (Phase Information)
P1 (Pig-Pen)
H1 (Hog-Pen)
I4 My Favorite Identifier
--- CURRENT STATES ---
S1 (water-jug)
--- CURRENT OPERATORS ---
O2 (fill)
O3 (fill)
```

So how do you delete items or save the list for future use? Next to the spinbox, is a menu button with the word "Edit" on it. Click on it and you will get a menu that will let you do one of three things: (1) save your list of custom WMEs to your tsDebugger prefs file, (2) discard the current list and reload the last saved one, and (3) delete the current selected item on the list.

Next to the menu button, is an auto-expand spinbox. Change the value on it, and all WMEs in the tree will auto-expand to that depth level.

WME Tree

This finally brings us to the actual WME tree. Note that there are two types of WMEs.

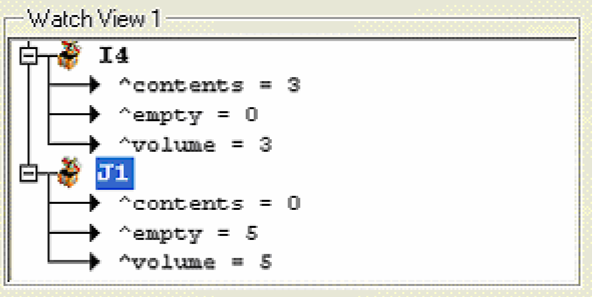
1. Some WME's point to other WMEs via a Soar Identifier, for e.g. the first item in the picture points to Soar identifier "I1". These items will have a gift box icon 🎁 next to them. Click on the plus sign to open the WME and see its children. The icon will change to an open gift box🎁.
2. Some WME contain simple data. These items will be pointed to by an arrow → rather than an icon.

Right-clicking on a WME

If you right-click on an identifier node in a true, the tsDebugger will present you with a context menu that allows you to perform certain functions on the display of that memory element.

- **Freeze item**: This will "freeze" the value of that item and all of its currently displayed children. In other words, the values of those attributes will not change in the display even if the values within the agent do change. The WME's icon will change to a gift box superimposed with an ice cube🧊. An annotation indicating the time that the WME was frozen will also be added, for e.g. "Frozen at time 17."
- **Unfreeze item**: This will allow the TSDebugger to resume updating of those WMEs contents. Note that the contents will update as soon as you hit unfreeze.
- **Send WME to Watch View X**: This will add the contents of that working memory element to one of the watch panes in the current layout. (more on this in the next section)

Watch Panes (4)



Watch View 1

```
I4
  ^contents = 3
  ^empty = 0
  ^volume = 3
J1
  ^contents = 0
  ^empty = 5
  ^volume = 5
```

Watch panes are very much like the WME tree. The only difference is that the WME tree has one root; the watch panes, on the other hand, are an arbitrary collection of WMEs. As explained above, to add an item to a watch pane, right click on a WME from either the main WME tree *or* another one of the watch panes and select the appropriate context menu item. To remove it, click on the item in a given watch pane and select "*Remove*."

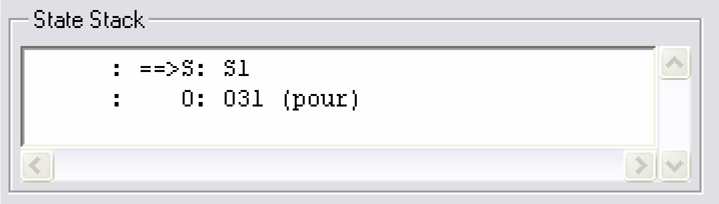
Some possible uses for watch panes:

- Organize different regions of WME interest without having to expand and scroll around the main tree. For example, you could have one watch view for your input link, one for scratch data structures it builds, one for important elaborations, etc.
- To keep an archive of previous values of certain WME's. Simply copy the item to a watch pane, freeze it, and see how it changes over time.

Miscellaneous Panes

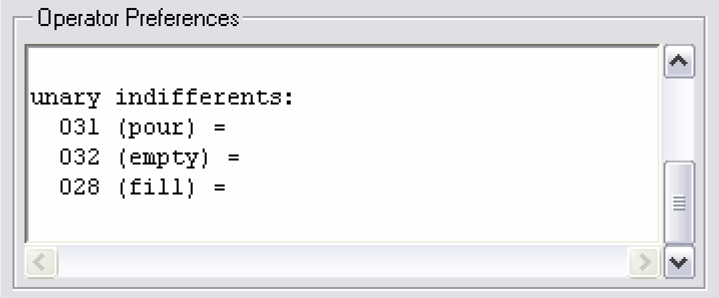
There are four miscellaneous panes that may be in a given layout:

- The state stack pane (5): shows the complete state stack of the agent.



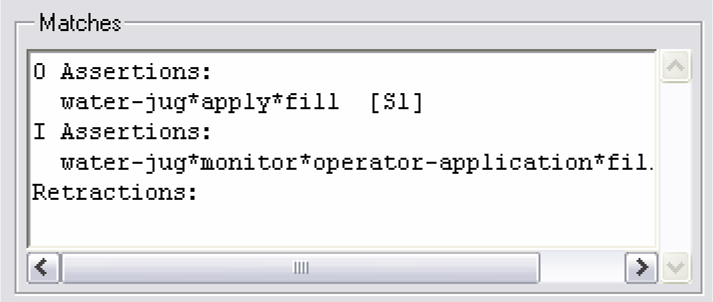
```
      : ==>S: S1
      :    O: O31 (pour)
```

- The operator preference pane (6): show all operator preferences in the lowest state of execution.



```

unary indifferents:
  O31 (pour) =
  O32 (empty) =
  O28 (fill) =
```

- The matches pane (7): shows the current operator matches and retractions in the lowest state of execution.

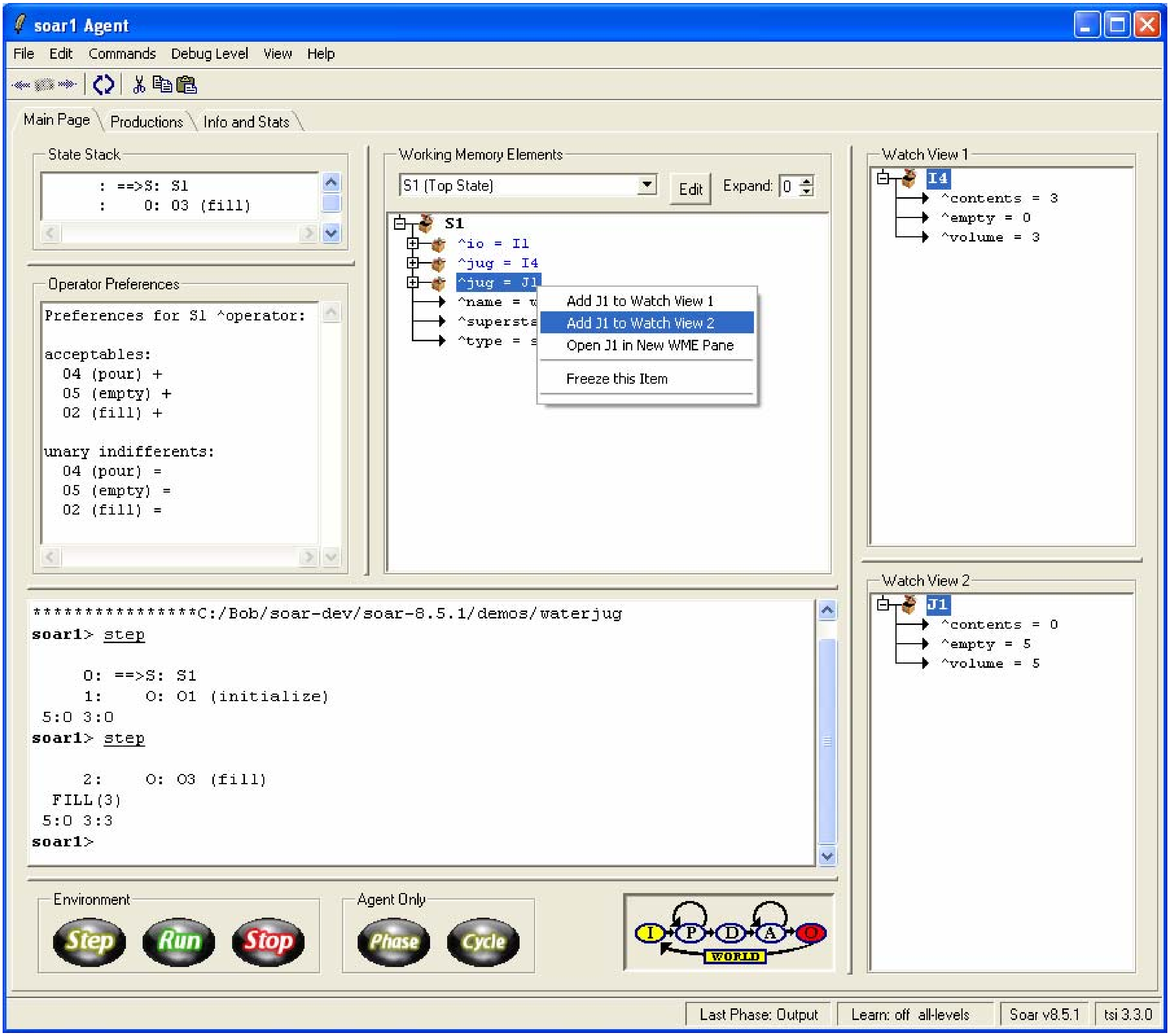


```
O Assertions:
  water-jug*apply*fill  [S1]
I Assertions:
  water-jug*monitor*operator-application*fil
Retractions:
```

More panes will be added in the future. Moreover, a rudimentary plug-in structure is implemented. No significant API will be available, so significant knowledge of TSDebugger code will be required.

Other Layouts

Default Layout



```
  : ==>S: S1
  :    O: O3 (fill)
```

```
Preferences for S1 ^operator:

acceptables:
  O4 (pour) +
  O5 (empty) +
  O2 (fill) +

unary indifferents:
  O4 (pour) =
  O5 (empty) =
  O2 (fill) =
```

```
S1
  ^io = I1
  ^jug = I4
  ^jug = J1
  ^name = w
  ^supersta
  ^type = s
```

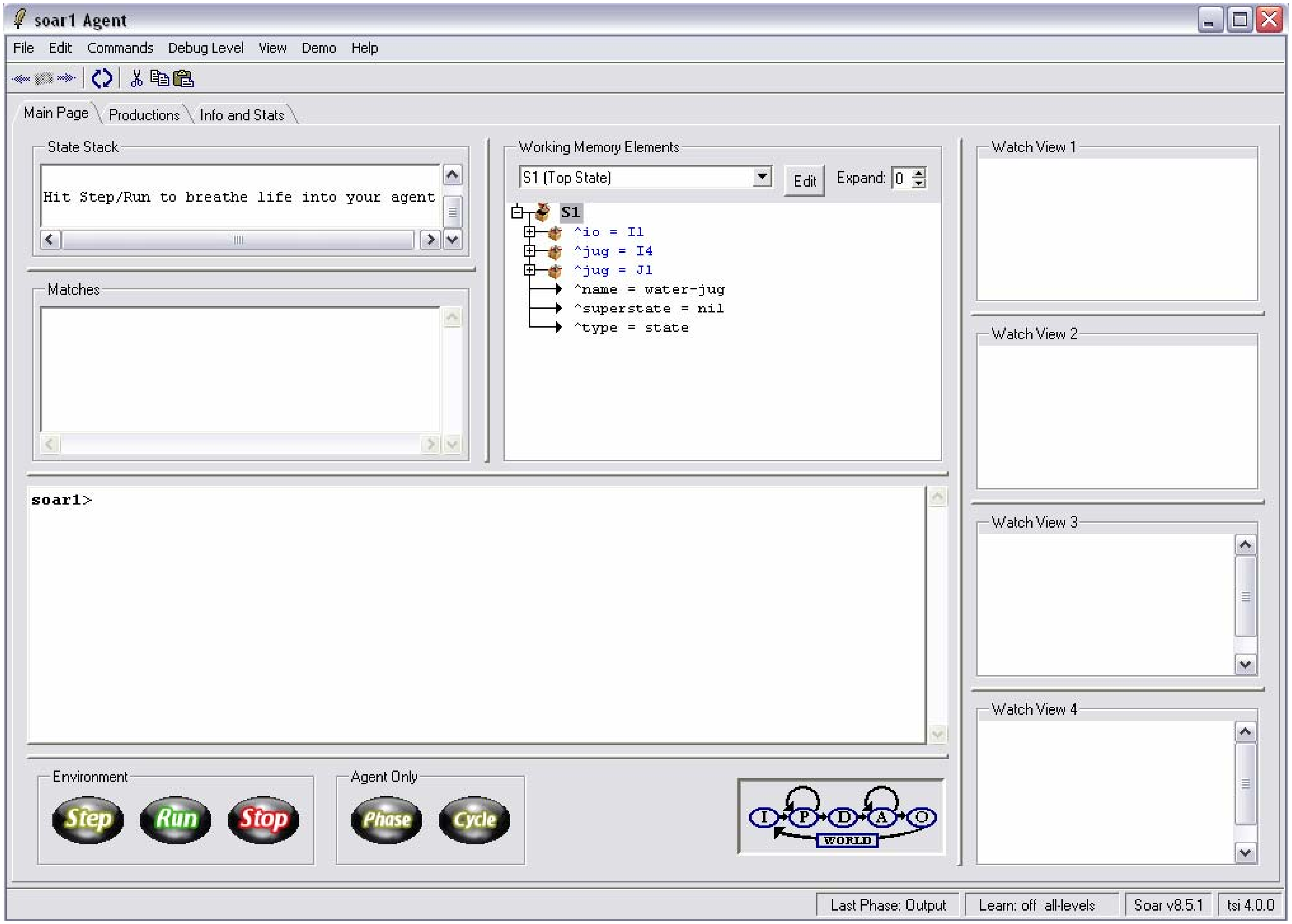
```
I4
  ^contents = 3
  ^empty = 0
  ^volume = 3
```

```
J1
  ^contents = 0
  ^empty = 5
  ^volume = 5
```

```
****************C:/Bob/soar-dev/soar-8.5.1/demos/waterjug
soar1> step

     0: ==>S: S1
     1:    O: O1 (initialize)
 5:0 3:0
soar1> step

     2:    O: O3 (fill)
  FILL(3)
 5:0 3:3
soar1>
```

The differences between the Default view and the Simple view are that the Default view has two Watch Views and it adds a pane which displays the current operator preferences. A minimum screen size of 1024x768 is recommended.

***More Watch Panes* Layout**



Hit Step/Run to breathe life into your agent

S1 (Top State)

Expand: 0

```
S1
  ^io = I1
  ^jug = I4
  ^jug = J1
  ^name = water-jug
  ^superstate = nil
  ^type = state
```

```
soar1>
```

Last Phase: Output

Learn: off all-levels

Soar v8.5.1

tsi 4.0.0

The other available view is the More Watch Panes view. This view has four Watch Views and replaces the Operator Preferences pane with a Matches pane, which shows the current production matches. A minimum screen size of 1280x1024 is recommended.

Note: the ability to create arbitrary layouts will be added to an upcoming version of the TSDebugger.

Menus

The other available menus are File, which allows you to source a soar file, load/save production memory, and manipulate the logging (i.e. turn logging on or off, etc). The Edit menu contains the standard Cut/Copy/Paste/Find and a Preferences dialog to customize the look of the TSDebugger. The Commands menu contains some items to clean up or reset and agent. The Debug Level menu allows one to fine tune the debugging output. The View menu, in addition to the options we’ve already discussed, also allows one to save or load the window preferences.