

SoarUnit

Bob Marinier
6/14/11



SOARTECH

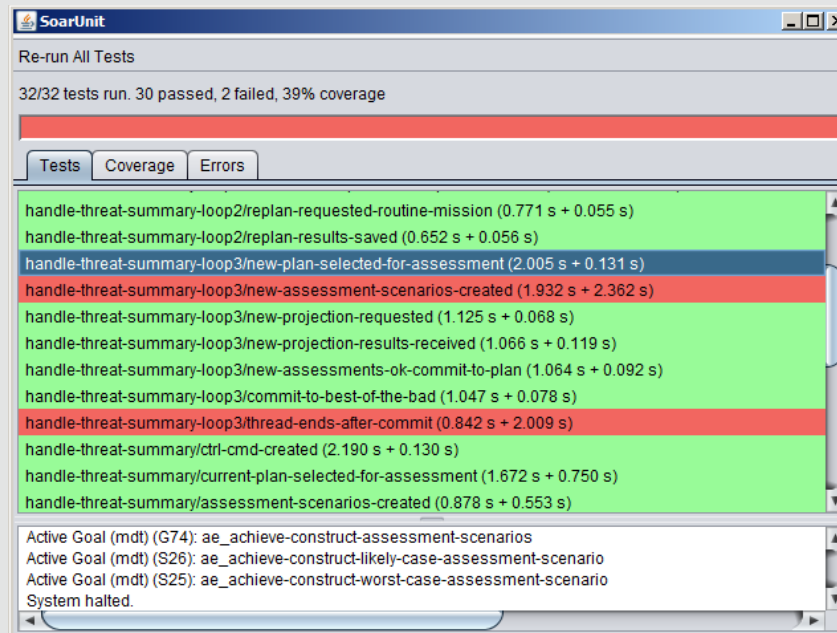
Modeling human reasoning.
Enhancing human performance.

Motivation

- Creating and maintaining Soar code is hard
 - Soar syntax/architecture does little to help (e.g., no type checking)
 - But partly this is what makes Soar powerful
- Some existing tools provide static help
 - Syntax checking/highlighting finds common syntactical issues
 - Datamap can find many structural issues
- But testing logic is still hard: need runtime testing
 - Some ad hoc solutions have been tried, but nothing particularly flexible or reusable
- Other languages have standard tools and frameworks for creating *unit tests*: small scale, repeatable tests that can be run quickly to test various parts of the system

SoarUnit

- Created by Dave Ray
- Inspired by JUnit (a unit testing framework for Java)
- Provides a way to define tests, including initial and success/failure conditions
- Part of JSoar (open source, BSD), but works with CSoar, too



Tests

- Define rules and commands to execute before the test runs
- One or more rules need to call the “pass” RHS function
- Can also call “fail”, but that’s uncommon
- Related tests in a single .soarunit file can share common rules and commands
- SoarUnit will process all .soarunit files in a given directory or directory tree

- A new Soar agent is created for each test, so no state is maintained across tests

Example

```
setup {
  source my-agent.soar
  learn --on
  sp {fake-input
    (state <s> ^io.input-link <il>)
    -->
    (<il> ^foo bar)
  }
}
```

```
test input-processing-started {
  sp {check-for-input-processing-operator-selected
    (state <s> ^operator <o>)
    (<o> ^name process-input)
    -->
    (pass)
  }
}
```

```
test input-processing-completed {
  sp {check-for-expected-structure
    (state <s> ^saw-foo-bar true)
    -->
    (pass)
  }
}
```

Other Features

- Can right-click on any test to launch debugger with that test loaded
- Coverage tab shows union of firing counts across all rules
- Error tab shown when there is a problem loading/running a test (e.g., syntax errors)
- In JSoar, supports Tcl and other scripting languages
- Can run multithreaded (multiple tests simultaneously)

The screenshot shows the SoarUnit application window. At the top, it says "Re-run All Tests" and "32/32 tests run. 30 passed, 2 failed, 39% coverage". Below this is a red progress bar. There are three tabs: "Tests", "Coverage", and "Errors". The "Tests" tab is selected, showing a list of rule names and their firing counts (FC). The list is as follows:

FC	Rule Name
0	top-state*propose*create-spatr-region
0	top-state*propose*modify-spatr-query*cgf-route-intersects*add-route
0	top-state*propose*modify-spatr-query*cgf-route-intersects*remove-route
0	top-state*propose*modify-spatr-query*cgf-waypoint-approaching*add-waypoint*fly-route
0	top-state*propose*modify-spatr-query*cgf-waypoint-approaching*add-waypoint*fly-to-waypoint
0	top-state*propose*modify-spatr-query*cgf-waypoint-approaching*remove-waypoint
0	top-state*propose*modify-spatr-query*cgf-waypoint-intersects*add-waypoint*fly-route
0	top-state*propose*modify-spatr-query*cgf-waypoint-intersects*add-waypoint*fly-to-waypoint
0	top-state*propose*modify-spatr-query*cgf-waypoint-intersects*remove-waypoint
0	utterance*processed*remove-from-queue
1	re_achieve-respond-to-assessments*infrastructure*propose*commit-to-current-plan
2	ngs*terminated-considerable*remove-on-timeout
2	refine-domain-primitive-action*apply*info-request*update-mission*ac-not-capable*TASKER
2	refine-domain-primitive-action*apply*info-request*update-mission-plan*ac-not-capable*TASKER
2	tasker_achieve-execute-cgf-plan-step*propose*send-command*update-mission
2	tasker_achieve-execute-cgf-plan-step*propose*send-command*update-mission-plan
2	tasker_convert-cmd-to-neutral-thread*apply*convert*update-mission
2	tasker_convert-cmd-to-neutral-thread*apply*convert*update-mission-plan
2	top-state*elaborate*goal*tasker_achieve-execute-cgf-plan-step*update-mission
2	top-state*elaborate*goal*tasker_achieve-execute-cgf-plan-step*update-mission-plan

Demo

Nuggets

- Has really helped me write complex agents
 - Helps identify bugs quickly
 - Supports agent development even when the environment is not ready
 - Helps spec out what the environment connection should look like

Coal

- Some missing features
 - Ability to re-run a single test or subset of tests
 - Ability to see the results of individual tests as they run
- Can be slow when using Tcl, or when the tests are failing
- A few minor bugs

More Info

- <https://code.google.com/p/jsoar/wiki/SoarUnit>