

Releasing pysoarlib and vim-soar-plugin

Aaron Mininger

University of Michigan

May 17, 2018

2018 Soar Workshop

Available Projects

Two personal projects available for use

- github.com/amininger/pysoarlib
 - Makes writing sml agents in python easier
- github.com/amininger/vim-soar-plugin
 - Support for editing soar files
 - Ability to run debugger inside vim

pysoarlib

pysoarlib - SoarAgent

Easy to run agents

```
from pysoarlib import *
agent = SoarAgent(
    agent_source="test-agent.soar",
    write_to_stdout=True)
agent.connect()
agent.execute_command("run 20")
agent.kill()
```

pysoarlib - AgentConnector

Abstraction for soar I/O

```
on_init_soar(self)
```

```
on_input_phase(self, input_link)
```

```
add_output_command(self, command_name)
```

```
on_output_event(self, command_name, root_id)
```

```
agent.add_connector("test", connector)
```

github.com/amininger/pysoarlib

pysoarlib - Identifier Extensions

Additional Methods on Identifier

```
(<id> ^legs 4 ^class chair ^class furniture)
```

```
id.GetChildInt("legs") -> 4
```

```
id.GetAllChildValues("class")
```

```
-> [ "chair", "furniture" ]
```

```
id.GetAllChildValues()
```

```
-> [ "4", "chair", "furniture" ]
```

vim-soar-plugin

vim-soar-plugin: Editing

Useful features for editing soar files:

- Proper Indenting
- Production aware copy/paste/delete
- Production templates (easy to add your own)

vim-soar-plugin: Debugging

Soar Debugging integrated into vim

- Unifies development and testing
- Allows you to write and edit while running
- Cursor-aware commands
(source, excise, print, matches)

vim-soar-plugin DEMO

Questions?