Releasing pysoarlib and vim-soar-plugin

Aaron Mininger
University of Michigan
Available Projects

Two personal projects available for use

- [github.com/amininger/pysoarlib](https://github.com/amininger/pysoarlib)
  - Makes writing sml agents in python easier

- [github.com/amininger/vim-soar-plugin](https://github.com/amininger/vim-soar-plugin)
  - Support for editing soar files
  - Ability to run debugger inside vim
Easy to run agents

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

github.com/amininger/pysoarlib
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
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" ]

github.com/amininger/pysoarlib
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)

github.com/amininger/vim-soar-plugin
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)

github.com/amininger/vim-soar-plugin
vim-soar-plugin
DEMO
Questions?