

Soar Workshop

SML Tutorial

Nate Derbinsky

While waiting... download Eclipse

- At least Java @ eclipse.org

Agenda

- Big picture
- System setup + Hello Soar
- Basic usage
- Additional resources

Big Picture: Soar Markup Language

- SML provides a programmatic interface into Soar based around sending and receiving commands packaged as XML packets. Used for...
 - Environments
 - Debuggers
 - Automated experimentation
- Written in C++
- Exported, via SWIG (swig.org) to...
 - Python
 - Java

System Setup (1)

1. Open Eclipse
2. New Java Project
 - Name=“MySMLProject”
 - Finish
3. New Class
 - Name=“MySMLClient”
 - Check: “public static void main...”
 - Finish

System Setup (2)

4. Inside main...

```
Kernel kernel;
```

5. Add `sml.jar` to class path

- Right click project -> Properties
- Java Build Path
- Libraries -> Add External Jar
- Locate `sml.jar` in `lib/soar/java`, Open, OK

6. Hover over “Kernel” (red underline)

- Click “Import ‘Kernel’ (sml)”

System Setup (3)

7. Finish main ...

```
Kernel kernel = Kernel.CreateKernelInNewThread();  
Agent agent = kernel.CreateAgent("soar");  
  
System.out.println(agent.ExecuteCommandLine("print s1"));  
  
kernel.Shutdown();
```

8. Run menu -> Run

System Setup (4)

9. Run menu -> Run Configurations

- Windows
 - Environment tab
 - New
 - Name=PATH
 - Value=c:\path\to\lib\soar
- Mac/Linux
 - Arguments, VM arguments
 - Djava.library.path=/path/to/lib/soar
- Run

Basic Usage

Part 1: Automation

- Command execution
- Loading rules
- Synchronous run control
- Capturing trace output

Part 2: Basic IO

- Managing WMEs on input-link
- Read output-link WMEs + feedback

Part 3: Event-Driven Environment

- Output handler

Command Execution

Syntax

“result”=agent.ExecuteCommandLine(“command”);

Try

- “stats”
- “epmem --stats”
- “sp {test (state <s> ^superstate nil) --> (<s> ^foo bar)}”
- “print test”

Loading Rules

Syntax

True/False = Agent.LoadProductions("location");

Try

1. Load: `path/to/file.soar`
2. Execute: "print"

Synchronous Run Control

Syntax

- Agent.RunSelf(numberSteps,stepSize = Decision);
- Agent.RunSelfForever();
- Agent.RunSelfTilOutput();
- Agent.ExecuteCommandLine(“run...”);

Try

1. Run: forever
2. Execute: “print --stack”

Capturing Trace Output

1. Create a PrintEventInterface (event handler)

```
public static final PrintEventInterface myPrinter = new PrintEventInterface() {  
    public void printEventHandler(int eventId, Object data, Agent agent, String message) {  
        System.out.println("Soar said: <" + message + ">");  
    }  
};
```

1. Register for Event

Syntax

```
Agent.RegisterForPrintEvent(eventId, handler, extraData);
```

Try

```
agent.RegisterForPrintEvent(smlPrintEventId.smlEVENT_PRINT, myPrinter, null);
```

Managing WMEs on `input-link`

Syntax

- `Identifier = Agent.GetInputLink();`
- `Identifier = Identifier.CreateIdWME("attr");`
- `FloatElement = Identifier.CreateFloatWME("attr", value);`
- `IntElement = Identifier.CreateIntWME("attr", value);`
- `WMEElement.DestroyWME();`

Try

```
agent.GetInputLink().CreateStringWME("foo", "bar");  
agent.RunSelf(1);  
System.out.println(  
    agent.ExecuteCommandLine("p --depth 3 s1"));
```

Read output-link WMEs + Feedback

Syntax

- `Int = Agent.GetNumberCommands();`
- `Identifier = Agent.GetCommand(Int);`
- `String = WMElement.GetAttribute();`
- `Int = Identifier.GetNumberChildren();`
- `WMElement = Identifier.GetChild(Int);`
- `WMElement = Identifier.FindByAttribute(String, Int)`
- `*Element = WMElement.ConvertTo*Element();`
- `Identifier.AddStatus<< Complete Error >>();`

Output Handler

Syntax

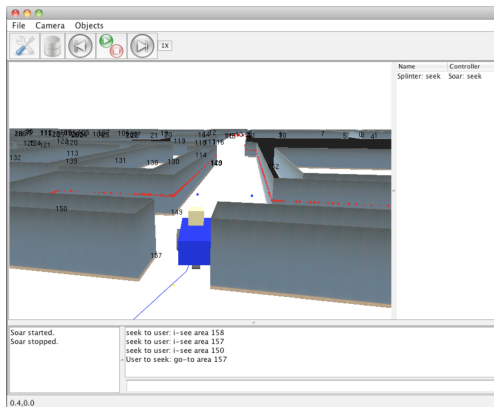
– Agent.AddOutputHandler(“cmd”, handler, data);

Try

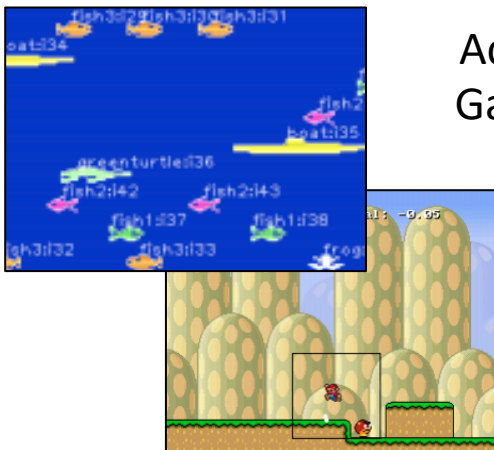
```
public static final OutputEventInterface cmdHandler = new OutputEventInterface() {
    public void outputEventHandler(Object data, String agentName, String attributeName, WMElement pWmeAdded) {
        for (int i=0; i<pWmeAdded.ConvertToIdentifier().GetNumberChildren(); i++) {
            final WMElement wChild = pWmeAdded.ConvertToIdentifier().GetChild(i);
            System.out.println("^" + wChild.GetAttribute() + " " + wChild.GetValueAsString());
        }
    }
};

agent.AddOutputHandler("do", cmdHandler, null);
agent.ExecuteCommandLine("sp {test (state <s> ^io.output-link <out>) --> (<out> ^do <d>) (<d> ^foo bar ^baz qux)}}");
agent.RunSelf(2);
```

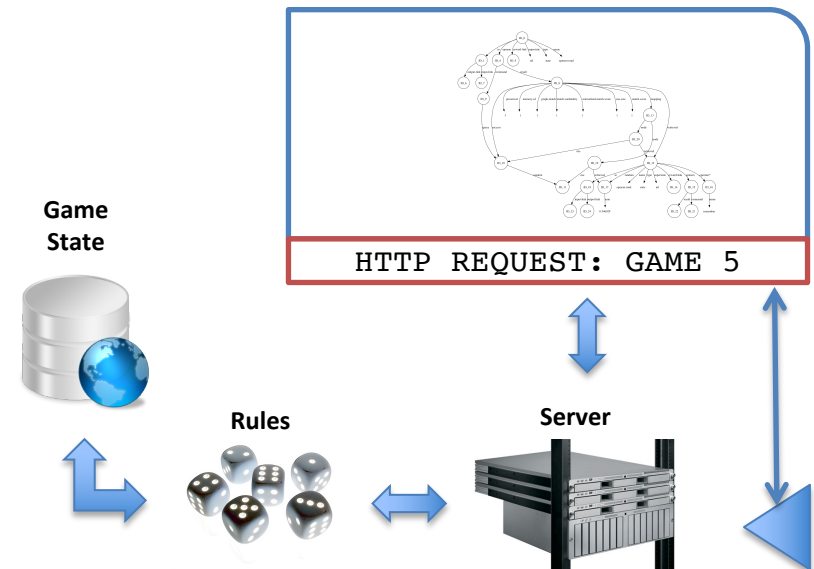
SML Example Environments



Cognitive Robotics

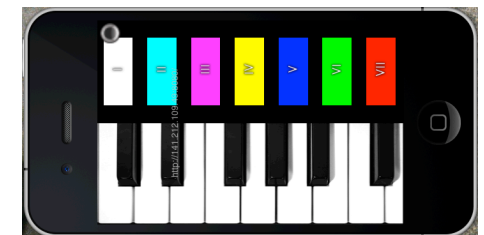


Action Games



Web Gaming

Interactive Mobile Music Generation



SimpleEaters

<https://github.com/SoarGroup/simple-eaters-world>

Additional Resources

- Quick Start Guide

<http://soar.eecs.umich.edu/articles/articles/soar-markup-language-sml/78-sml-quick-start-guide>

- Threads in SML

<http://soar.eecs.umich.edu/articles/articles/soar-markup-language-sml/203-threads-in-sml>

- Compiling SML Clients

<http://soar.eecs.umich.edu/articles/articles/soar-markup-language-sml/79-how-to-compile-sml-clients>