

SML Tutorial

Soar Workshop 31 – Nate Derbinsky

While waiting...

1. Make sure you have internet access

2. Download Soar 9.3.1

soar.googlecode.com

3. Download Eclipse (with at least Java)

www.eclipse.org

4. Download tutorial support files

www.eecs.umich.edu/~nlderbin/workshop31

Agenda

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

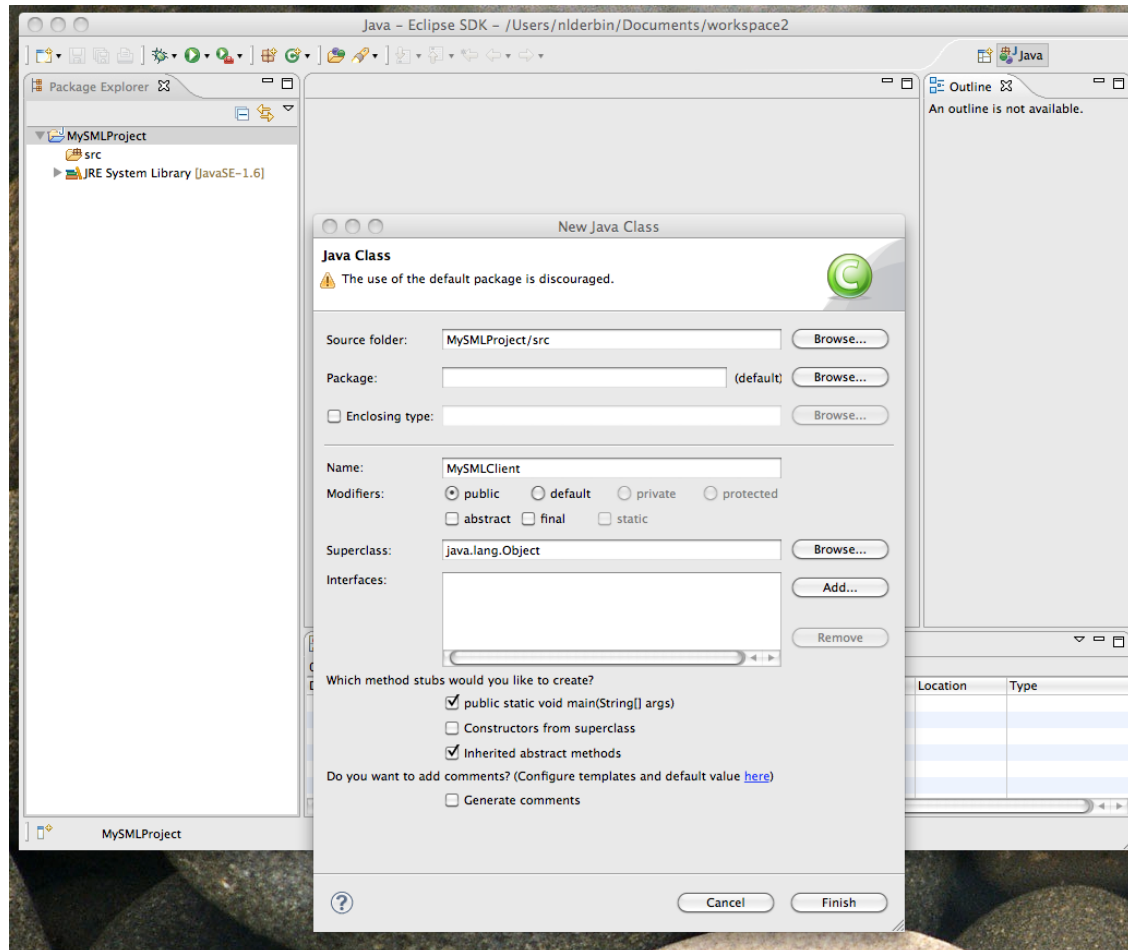
Big Picture: Soar Markup Language

- 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

Status



System Setup (2)

4. Inside main...

```
Kernel kernel;
```

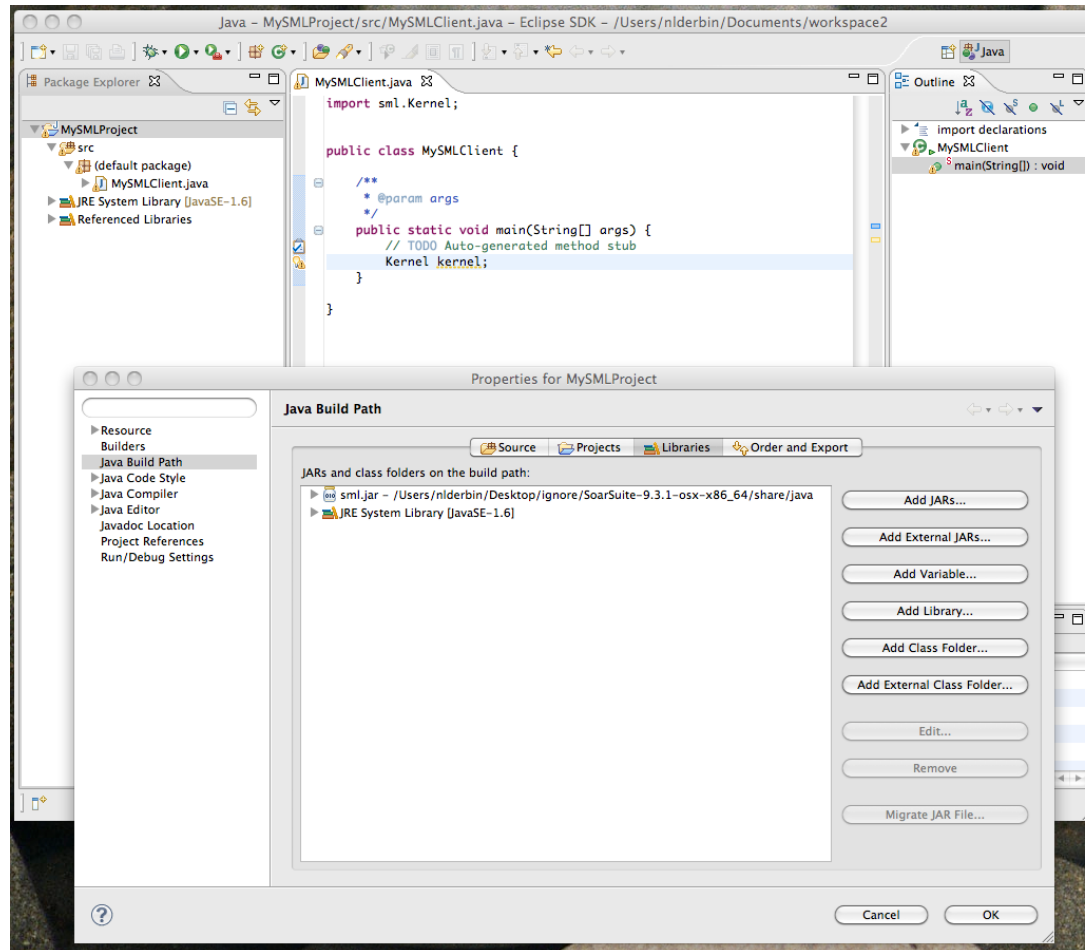
5. Add `sml.jar` to class path

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

6. Hover over “Kernel” (red underline)

- Click “Import ‘Kernel’ (sml)”

Status



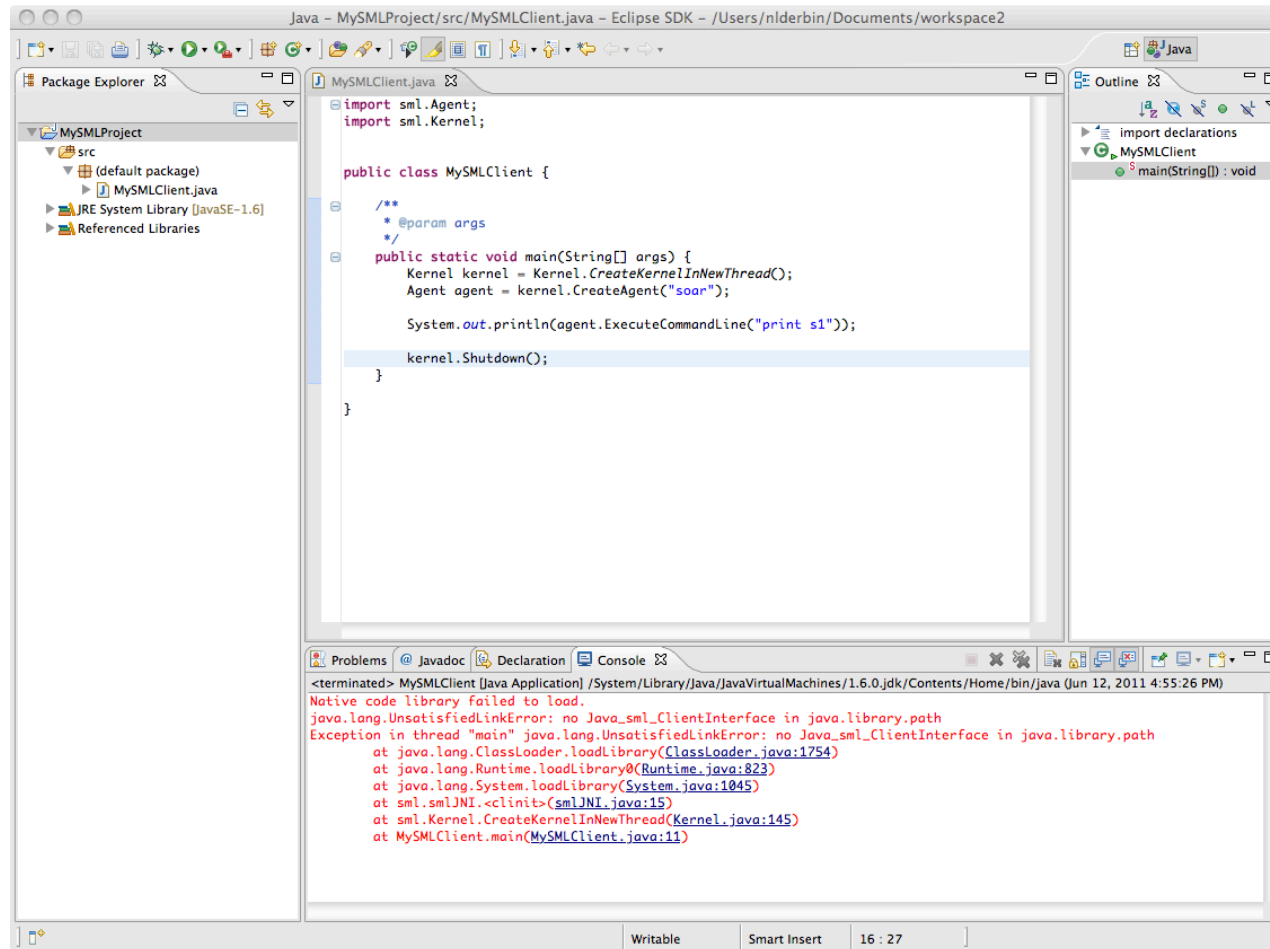
System Setup (3)

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

Status

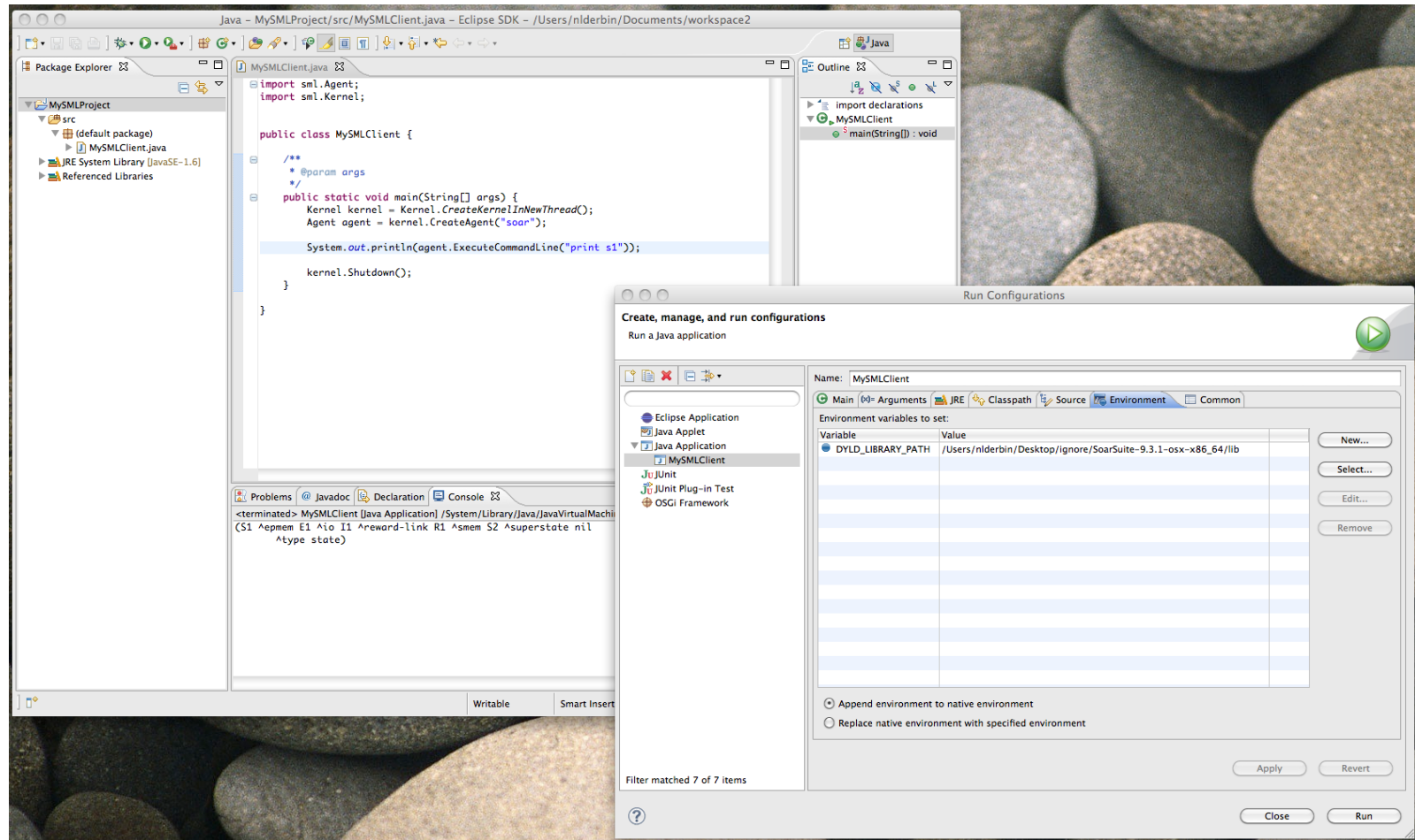


System Setup (4)

9. Run menu -> Run Configurations

- Environment tab
- New
 - Name=
 - Mac: DYLD_LIBRARY_PATH
 - Linux: LD_LIBRARY_PATH
 - Windows: PATH
 - Value="/path/to/soar/release/" +
 - Mac/Linux: "lib"
 - Windows: "bin"
 - Run

Status



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: goodbye . soar
2. Execute: "print goodbye"

Synchronous Run Control

Syntax

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

Try

1. Load: goodbye.soar
2. Run: forever
3. Execute: “print s1”

Capturing Trace Output

print.java

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 + ">");  
    }  
};
```

2. Register for Event

Syntax

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

Try

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

Managing WMEs on `input-link`

input.java

Syntax

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

Task. Add and remove WMEs of differently typed values to the input-link. Use execution and run-control to verify via System printing.

Read output-link WMEs + Feedback

output.java

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 >>();`

Task. Have an agent produce output. Parse via SML and provide feedback to the agent. Verify via agent action and working memory inspection.

Output Handler

handler.java

Syntax

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

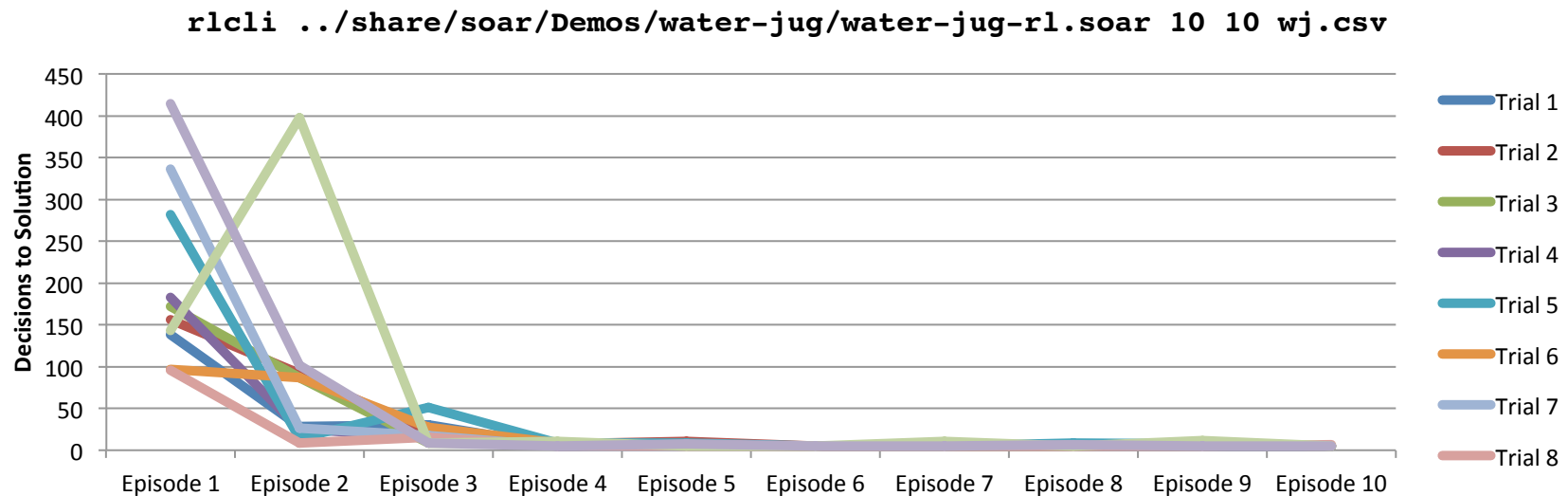
Task. Choose a secret number. Have a Soar agent guess the value via output commands until correct.

SML Example Applications

Automation

RLcli

Automates collection of RL-related run data over variable agents/episodes/trials

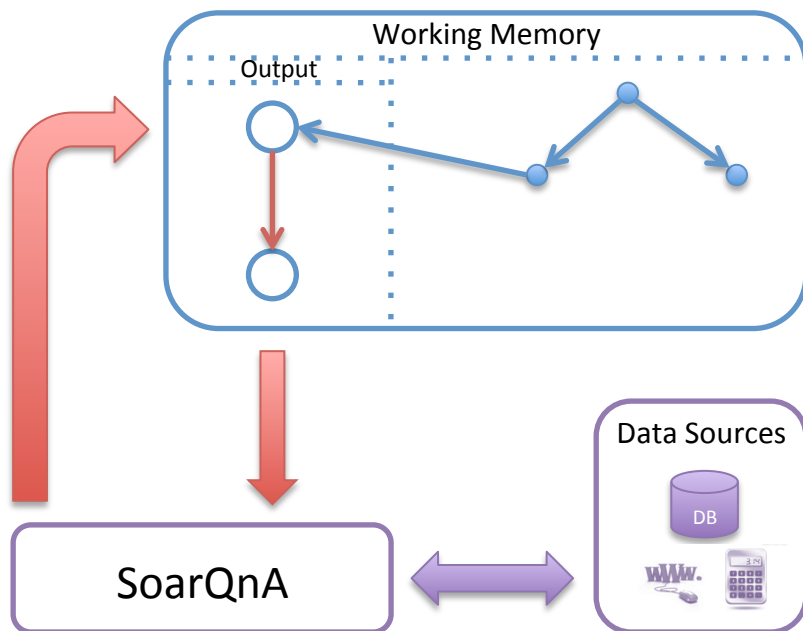


SML Example Applications

Extending Soar's I/O

SoarQnA

Standardized access to external knowledge



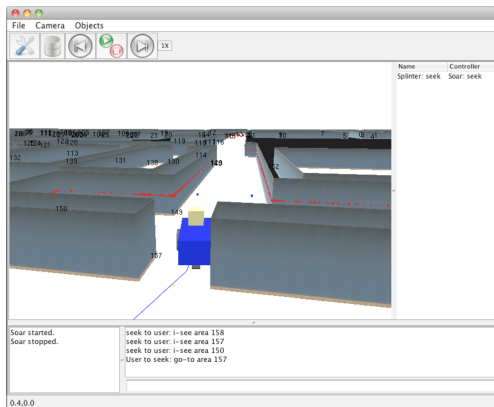
```
Soar Debugger in Java - soar1

print --depth 20 i3

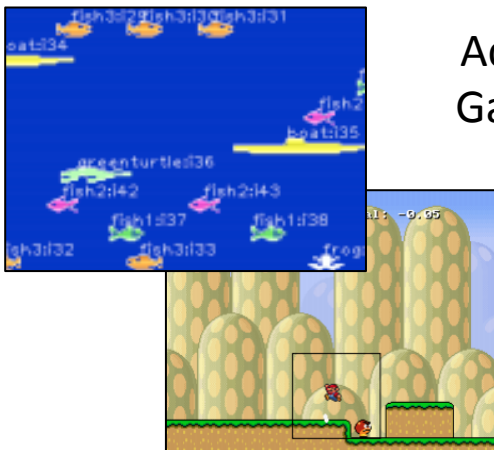
(i3 ^qna-query Q2)
(Q2 ^id 1 ^parameters P1 ^query imdb ^result R4 ^results incremental
 ^source http-json-get ^status complete)
(P1 ^i 2011 ^t1 Midnight in Paris)
(R4 ^features F1 ^next nil ^num 1)
(F1 ^actors IOwen Wilson, Rachel McAdams, Kathy Bates, Kurt Fuller
 ^director IWoody Allen! ^genre IComedy, Fantasy, Romance
 ^ID tt1605783
 ^Plot IA romantic comedy about a family traveling to the French capital for business. The party includes a young engaged couple
 ^Poster Ihttp://ia.media-imdb.com/images/M/MV5BMTM4NjY1MDQwMl5BMl5BanBnXkFtZTcwNTI3Njg3NA@@_V1_SX320.jpg!
 ^Rated PG-13 ^Rating I7.8! ^Released I11 May 2011! ^Response True
 ^Runtime I1 hr 40 mins! ^Title IMidnight in Paris! ^Votes I1352!
 ^Writer IWoolly Allen! ^Year I2011!)
```

SML Example Applications

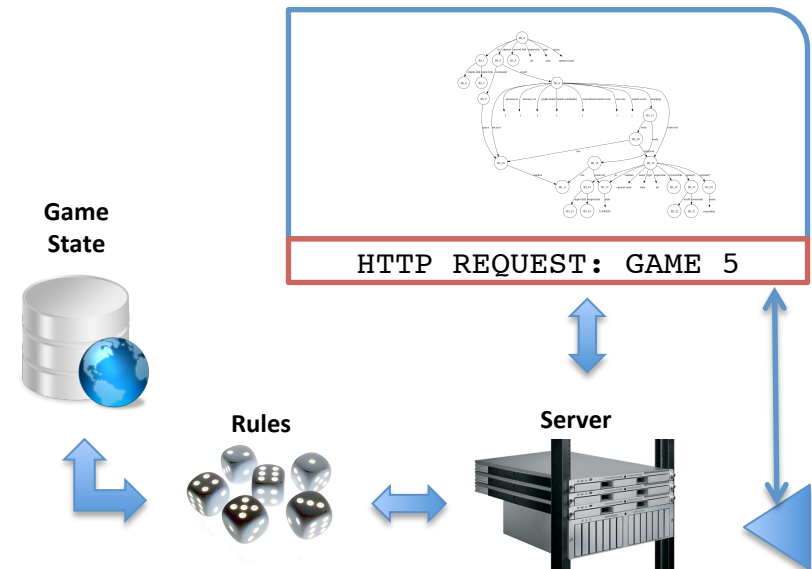
Environments



Cognitive Robotics

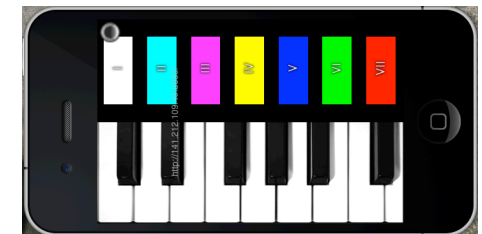


Action Games



Web Gaming

Interactive Mobile Music Generation



Additional Resources

- Quick Start Guide
<http://code.google.com/p/soar/wiki/SMLQuickStartGuide>
- Threads in SML
<http://code.google.com/p/soar/wiki/ThreadsInSML>
- Simple Environment Examples
<http://code.google.com/p/soar/wiki/HelloWorld>
- SoarScratchPad
[trunk/SoarSuite/SoarScratchPad](#)

Extra Time

- Create an SML wrapper to the WordNet demo. Allow a user to input the word/POS pair and have Soar search WordNet for sense definitions, which are shown to the user.
- Allow the user to input a sequence of strings, supplied to Soar via input-link. Use episodic memory to repeat the sequence verbatim back to the user via output.