#### **Soar Data Collection**

Jonathan Voigt 31<sup>st</sup> Soar Workshop June 2011



Modeling human reasoning. Enhancing human performance.

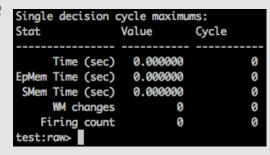
### **Motivation**

- Stats accumulates values but not history
- Stats history useful for agent evaluation
  - What is the worst-case reactivity?
- Optimizing agents for long-running robot experiments

	Input	Propose	Decide	Apply	Output	Computed Totals			
ernel:	0.000	0.000	0.003	2.909	0.000	2.912			
input fn:	0.000				 !	0.000			
utpt fn:					0.000	0.000			
allbcks:	0.000	0.000	0.000	0.000	0.000	0.000			
omputed									
	0.000	0.000	0.003	2.909	0.000	2.912			
alues from s	ingle t	imers:							
Kernel CPU T		2.916	sec.						
Total CPU T	ime:	2.920	sec.						
060 decision	s (0.95	3 msec/dec	ision)						
9239 elaboration cycles (3.019 ec's per dc, 0.316 msec/ec)									
9240 inner elaboration cycles									

## **Approach**

- Kernel changes to stats command: "max-cycle" stats collection
  - Time, production firings, working memory size
  - EpMem/SMem time each cycle
  - Efficient history storage



- External tool: Implemented in Java in SML
  - Idea: Flexible for easy integration with existing environments
- Focus on performance
  - Minimize expensive API calls, "uncertainty principle"
- Collect data rows in periods of decision cycles or elapsed time
- Output to spreadsheet-compatible format for easy inspection
  - CSV (comma separated values)

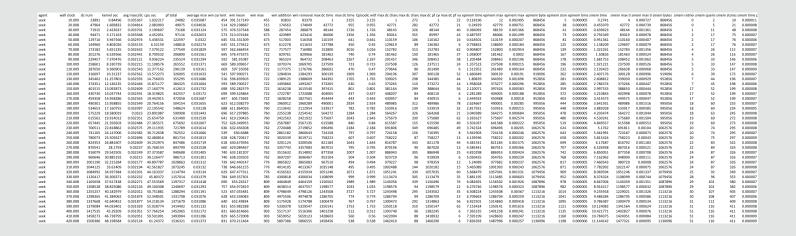
#### **Data**

- Agent name and settings (e.g. what learning systems are on)
- Wall clock, Kernel times, CPU times
- Production firing counts and firing time
- Working memory element counts: max, mean, min, additions, removals
- Episodic and semantic memory: time, retrievals, queries, stores etc.
- Many statistics record maximums and means that reset on each call to collect
  - "How long did the slowest decision cycle take since I last checked?"

### **Example Data**

agent	wall clock	dc num	kernel sec	avg msec/dc	cpu sec	pf total	average mse	wm current	wm mean	wm max
seek	10.009	23891	0.84496	0.035367	1.032217	24082	0.035087	454	391.517149	455

wm addition: wm removal: max dc time max dc time Episodic with max dc chan; max dc chan; max dc pf cy max dc pf val 83833 83379 1 3155 3.155 1 272 1 22



### **Usage Details**

- Single class Java library
- Instantiate once for all agents
- Call methods during three events:
  - Soar system start: onStart
  - Update (after all agents pass output): onUpdateEvent, collect
  - Soar system stop: onStop, collect
- Utility methods to help timing collection
  - Every n decision cycles: setPeriodCycles
  - Every n milliseconds: setPeriodMillis
- Flushing to file is expensive, doesn't happen while agents are running
  - Should explicitly call every so often in case of catastrophic failures

# **Nuggets/Coal**

- Lots of work to
  - Support different output formats: sqlite, speedy
- Tricky to use because of the performance goal
- Successfully used to collect data during long runs
- Useful to optimize agents

### **Download**

- Currently on github
  - https://github.com/voigtjr/soar-datacollector
- Will be moved in to SoarSuite