# 2014 Soar Releases

Mazin Assanie
University of Michigan
mazina@umich.edu



# Schedule of Releases

- Soar 9.3.3
  - Two years of bug fixes and enhancements
  - Key new features: CDPS, TclSoarLib
  - Performance improvements
- Soar 9.4
  - Key new feature: SVS
- Soar 9.5 Beta
  - Key new feature: Generalized chunking

Available now

Parallel Releases July/Aug

# Soar 9.3.3

- Available now.
- Two years of bug fixes and enhancements
  - Better feedback and more consistent output
  - See full release notes
- Two big new features

# Chunks Can Utilize More Knowledge

- Chunking can now incorporate knowledge about **WHY** an operator was selected in a substate.
- This additional knowledge means that your chunks can become *more specific*.

# Lego Robot Example

- Robot doesn't know how to move so enters substate
- Proposes three operators
  - Turn left, turn right, go forward
- Search control knowledge
  - If I see black, prefer forward operator
  - If I see white, prefer left operator
  - If I see brown, prefer right operator

# Lego Robot Example

```
sp {propose-move-forward
    (state <s> ^superstate <ss>)
    (<ss> superstate nil)
-->
    (<s> ^operator <o>)
    (<o> ^name move-forward)
}

sp {apply-move-forward
    (state <s> ^operator <o>
               ^superstate.io <io>)
    (<o> ^name move-forward)
-->
   (<io> ^output-link <out>)
   (<out> ^move forward)
}
```

```
sp {search-control-black
   (state <s> ^superstate <ss>
              ^operator <o1> +
              ^operator <o2> +)
   (<o1> ^name move-forward)
   (<ss> ^io <io>)
   (<io> ^input-link <in>)
   (<in> ^sensor <sensor>)
   (<sensor> ^color black)
-->
   (<s> ^operator <o1> > <o2>)
}
```

# Lego Robot Example

```
sp {chunk-9.3.2
   (state <s> ^superstate nil
              ^io <io>)
-->
   (<io> ^output-link <out>)
   (<out> ^move forward)
}
```

```
sp {chunk-9.5
   (state <s> ^superstate nil
              ^io <io>)
   (<io> ^input-link <in>)
   (<in> ^sensor <sensor>)
   (<sensor> ^color red)
-->
   (<io> ^output-link <out>)
   (<out> ^move forward)
}
```

# Lego Robot Example

```
sp {chunk-9.3.2
   (state <s> ^superstate nil
              ^io <io>)
-->
   (<io> ^output-link <out>)
   (<out> ^move forward)
}
```

```
sp {chunk-9.5
   (state <s> ^superstate nil
              ^io <io>)
   (<io> ^input-link <in>)
   (<in> ^sensor <sensor>)
   (<sensor> ^color red)
-->
   (<io> ^output-link <out>)
   (<out> ^move forward)
}
```

# Chunking Search Control Knowledge

- Allows you to use chunking with agents that would previously produce over-general chunks.
- Off by default. You can turn on with the commands:

```
learn --desirability-prefs
```

or

```
learn -p
```

# TclSoarLib

- Seamlessly turns any Soar prompt into a Tcl prompt with a new command:

```
cli tcl on
```

- Users can use Tcl variables and functions directly within their soar code or make Tcl function calls on the RHS of rules.


- This component was funded by Soar Technology.

# TclSoarLib Uses

- Defining constants
  - Saves time
  - tcl catches errors
  - Easy to change
- Automatic generation of complex or tedious rules or conditions
- Running arbitrary tcl code to debug or log agent behavior

# TclSoarLib Uses

- **New Goal System** (https://code.google.com/p/new-goal-system/)
  - A Soar library to support faster development of Soar agents by allowing programming at a higher level of abstraction.
  - Through Tcl macros, NGS incorporates elements of encapsulation to hide many of the low-level Soar details of object creation.
  - Provides interfaces for matching to standard object structures such as goals and operators.
  - Forest of Goals: Ability to have multiple active goals at once
  - Actions from different active goals are interleaved by default

# Tcl Example

```
sp {initialize*state*directions
   (state <ss> ^type state)
   -->
   (<ss> ^directions <n> <e> <s> <w> <ne> <se> <nw> <sw>)
   (<n> ^value north ^opposite south)
   (<s> ^value south ^opposite north)
   (<e> ^value east  ^opposite west)
   (<w> ^value west  ^opposite east)
   (<ne> ^value north-east ^opposite south-west )
   (<se> ^value south-west ^opposite north-east )
   (<nw> ^value north-west ^opposite south-east )
   (<sw> ^value south-east ^opposite north-west )
}
```

# Tcl Example

```
proc defOppDirs { dir1 dir2 } {
    sp "initialize*state*directions*$dir1*$dir2
        (state <ss> ^type state)
    -->
        (<ss> ^directions <dir1> <dir2>)
        (<dir1> ^value $dir1 ^opposite $dir2)
        (<dir2> ^value $dir2 ^opposite $dir1)“}

defOppDirs north south
defOppDirs east west
defOppDirs north-east south-west
defOppDirs north-west south-east
```

# Soar 9.4

- Available July 1st – Aug 1st 2014.
- Soar Visual System (SVS)
  - Spatial reasoning component
  - Next talk will be about SVS
- More bug fixes and enhancements that were not ready for 9.3.3 release

# Soar 9.5 Beta

- Available July 1st – Aug 1st 2014
- Has been in parallel development for over a year
- Main new feature is generalized chunking
- Changes to several core data structures, significant refactoring and code clean up
- User interface enhancements

# Generalized Chunking

- Can variablize anything, not just short-term identifiers.
  - Utilizes previously unused information from original productions
  - Performs an analysis of the identity of symbols used in problem-solving

# Generalized Chunking

- Variables in chunks can now have complex constraints placed on them
  - Determined by the cumulative constraints on the variable from all rules that led to that chunk being formed

# Nuggets

- Many exciting new capabilities
  - Users can use chunking in situations where they could not before.
  - Users can be more confident about using chunking.
  - Users can integrate spatial reasoning into their agents more easily.
  - Existing systems are maturing and becoming more stable and feature-rich.
- Easier to use

# Nuggets

- Full functionality of 9.5 completed
  - Currently debugging, improving efficiency and polishing
- Performance of 9.3.3 (preliminary)
  - 33% faster for a normal optimized build

# Coals

- Performance of 9.3.3 (preliminary)
  - 5% slower with high-performance no timer build
- Existing episodic and semantic memory databases from 9.3.2 and below will not load in 9.3.3.
  - Conversion tool will be available in 9.4.
- New append-database option can delete databases.