

# Soar Command Line Interface

## Soar 8.6

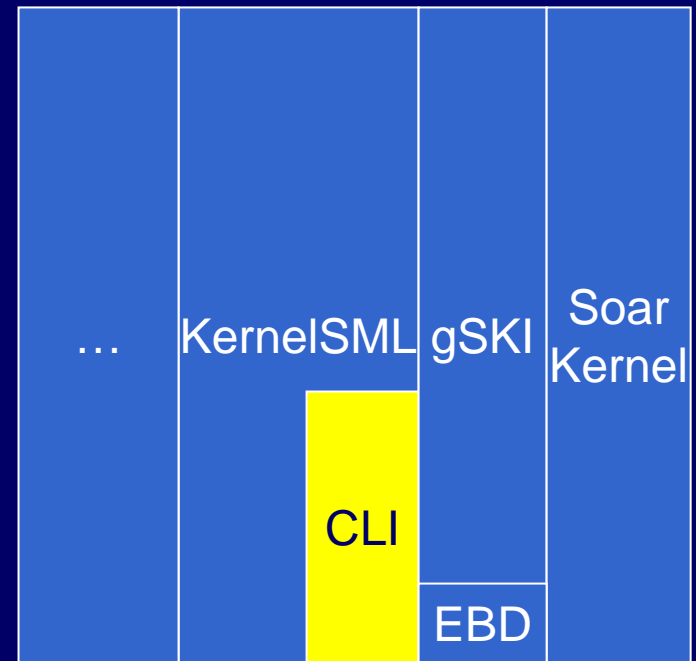
Jonathan Voigt  
University of Michigan  
Soar Workshop 25

# Problems with 8.5's Command Line

- *Soar 8.5's command line syntax*
  - Sometimes undefined
  - Occasionally inconsistent
  - Difficult for new users
- *Tcl legacy*
  - Although the Soar C interface is independent of Tcl, the same is not true for all of the commands
    - Some commands implemented in the TSI or wrapped by the TSI for full functionality
    - Some "Soar" commands are Tcl commands
      - source
- *Order of options & arguments*
  - Often important when it could be irrelevant
- *Client-side implementation*

# A Solution: The Soar 8.6 Command Line Interface (CLI) Module

- Managed by KernelSML (server-side)
- Communicates with kernel using gSKI
  - EvilBackDoor (EBD) used for required features incomplete or unimplemented in gSKI
- Implements a revised command syntax



# Soar 8.6 CLI Syntax Goals

- *Consistency*
  - Use the same syntax patterns across the space of commands
  - Easy to learn
- *Flexibility*
  - Many different ways to do the same thing
- *Legacy*
  - Familiarity to existing Soar users

# Syntax Changes

- New syntax based on GNU GetOpt
  - Part of GNU C library
    - [http://www.gnu.org/software/libc/manual/html\\_node/Getopt.html](http://www.gnu.org/software/libc/manual/html_node/Getopt.html)
- Widely used for command line option and argument processing
- Open source strikes back: GetOpt code taken and modified for use in Soar 8.6 CLI

# Soar 8.6 Arguments

- **Argument:** An element of the command line separated by white space
- **Command name:** The first argument

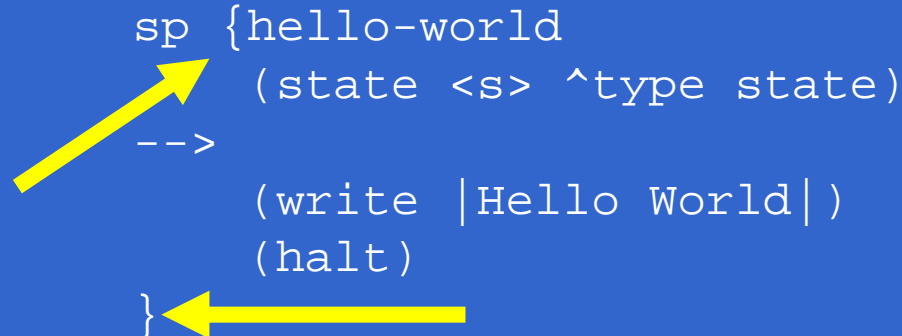
• White space inside quotes, braces and parenthesis **ignored**

• 4 arguments:

```
watch -l 2 -r
```

• 2 arguments:

```
sp {hello-world
    (state <s> ^type state)
    -->
    (write |Hello World|)
    (halt)
}
```



# Soar 8.6 Options

- **Short options:** Arguments prefixed by a single dash

`-c, -O, -cPp`

Three options!

- Short options may be combined with one dash
- **Long options:** Arguments prefixed by two dashes

`--chunks, --disable`

# Soar 8.6 Option Arguments

- **Option argument:** An argument immediately following an option that takes an argument

4 is the option argument:

```
watch --level 4
```

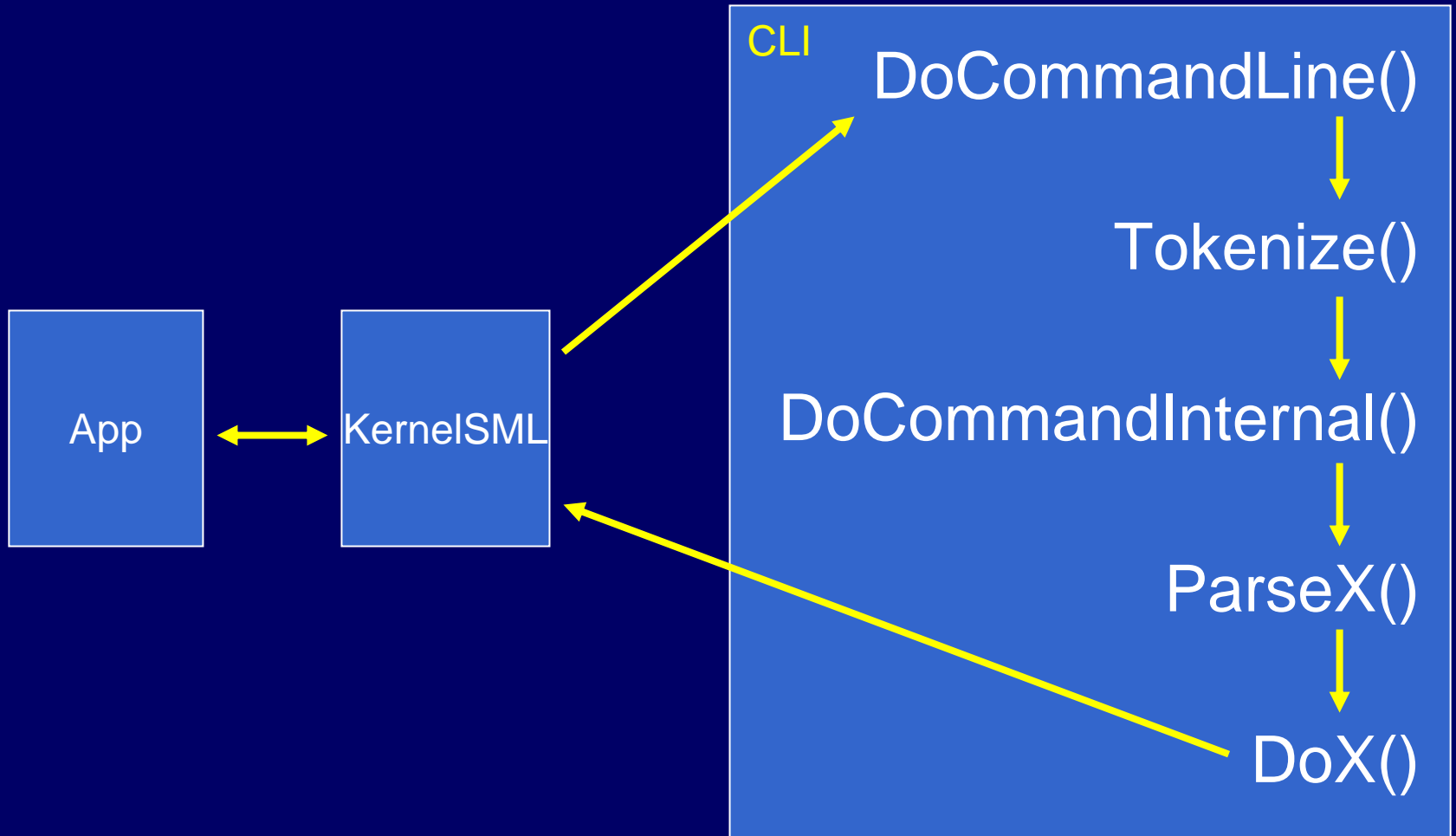
“removes” is the option argument

```
watch-wmes --add-filter -t removes S3 * *
```

- Option arguments **may be optional**
  - Optional option arguments cannot start with a dash!



# The Life of a Command Line



# DoCommandLine()

- Called from KernelSML
  - Part of the exposed interface
- **Handles logging**
  - Top-level
  - Command line is 'pure'
- **Handles communication with KernelSML**
  - Marshals XML command result

# Tokenize()

- First stage of parsing
- **Splits command line** into a vector of *arguments*
  - Splits using spaces, preserving spaces between braces, parenthesis and quotes
- **Removes comments**
  - Pound sign (#) comments often found inside Soar files

# DoCommandInternal()

- **Help**

- Looks for help options on the line
- Calls help command if found

```
watch -help  
excise -h
```

- **Aliases**

- If command name is an alias, substitutions are performed on the argument vector

```
x -a → excise -a  
d → run -d 1
```

- **Partial matches**

- If command name is not an alias, it is checked against all commands to see if there is a match
- Substitutes matched command name if found

```
exci → excise
```

# ParseX()

- Parsing separated from processing
  - Easier to maintain
- **Parses the data** required to process the command out of the argument vector
  - Options and option arguments in all commands parsed out using GetOpt
  - Remaining arguments dealt with according to command syntax
- All ParseX() function signatures are the same

```
ParseWatch(gSKI::IAgent* pAgent, std::vector<std::string>& argv)
```

# Option Ordering

- No processing is done until command is parsed
  - An exception: add-wme (WME parsed by EvilBackDoor)
- Options can be in any order
  - Options processed left to right
  - If options conflict, the last option wins

```
watch -r -1 2 ...preferences turned on, then off by -1 2  
watch -1 2 -r ...intended behavior (level 2 with preferences)
```

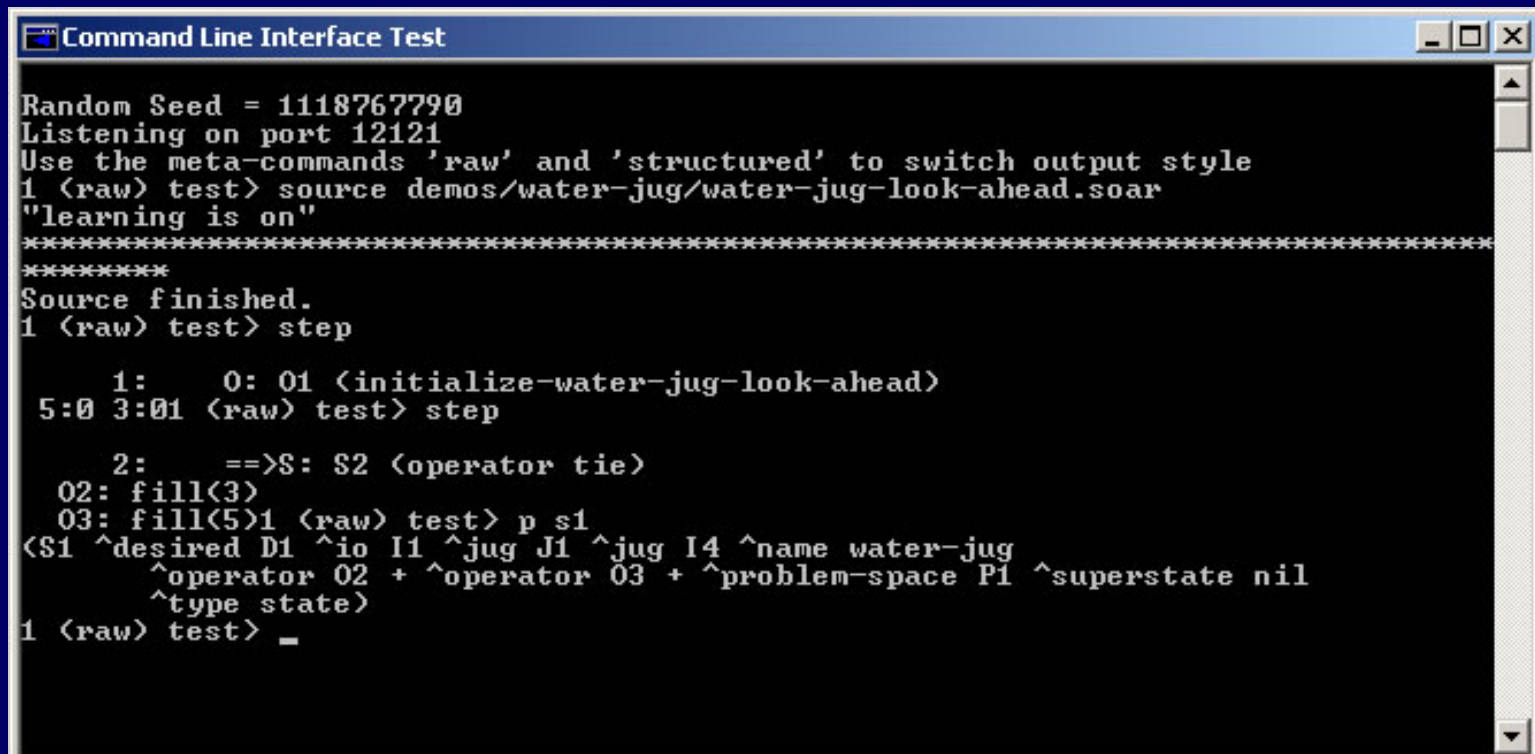
# DoX()

- Does the actual processing required to execute the command
- Most DoX() function signatures are different
  - Depends on the data required to execute the command

```
DoWatch(gSKI::IAgent* pAgent, const WatchBitset& options,  
        const WatchBitset& settings, const int wmeSetting,  
        const int learnSetting)
```

# Command Line Interface Test App

- Included with Soar 8.6 is a simple CLI module test application
- Easy to debug simple changes to the command line interface inside Visual Studio using this app



```
Command Line Interface Test
Random Seed = 1118767790
Listening on port 12121
Use the meta-commands 'raw' and 'structured' to switch output style
1 (raw) test> source demos/water-jug/water-jug-look-ahead.soar
"learning is on"
*****
*****
Source finished.
1 (raw) test> step

      1:      0: 01 (initialize-water-jug-look-ahead)
5:0 3:01 (raw) test> step

      2:      ==>S: S2 (operator tie)
      02: fill(3)
      03: fill(5)1 (raw) test> p s1
(S1 ^desired D1 ^io I1 ^jug J1 ^jug I4 ^name water-jug
      ^operator O2 + ^operator O3 + ^problem-space P1 ^superstate nil
      ^type state)
1 (raw) test> _
```



# Nuggets

- **Syntax consistency**
  - Standard option parsing
- **Simple maintenance**
  - Parsing separated from processing
  - Well documented
- **Flexible option ordering**
  - Last option wins
- **Same command interface for all clients**

# Coals

- **Many syntax changes**
  - Bad for existing users
- **EvilBackDoor and old interface usage**
- **Some command output generated by kernel**
  - via print callbacks
- **Structured (XML) output incomplete**

# Questions?

- Online resources:
  - [voigtjr@gmail.com](mailto:voigtjr@gmail.com)
  - [soar-group@lists.sourceforge.net](mailto:soar-group@lists.sourceforge.net)
  - <http://sitemaker.umich.edu/soar>
  - <http://winter.eecs.umich.edu/soarwiki>