The Soar Blank Environment

A Skeletal System for Building Tcl-based Soar Environments

Mazin Assanie

University of Michigan

Artificial Intelligence Lab

Motivation

- Reuse previous efforts in environment creation
  - 40% of Tanksoar code
- Decrease knowledge required to start using Soar and/or creating testbeds
  - Hide mechanics of advanced Soar usage
  - Hide tricker Tcl subjects like multiple interpreters, aliases, etc.
  - Users spend more time on what they're building rather than learning the intricacies of I/O and putting Soar and Tcl together

Advantages

- Tcl/Tk and Soar self-contained
- TSI properly configured to be used with an environment
- Event loop, agent setup and common environment functions already implemented
- Utility functions for managing Soar working memory defined

General How You Use

- Determine the rules of the environment
- Determine your agent percepts, i.e. the structure of the input-link
- Specify valid output commands
- Add code to the SBE in the appropriate places
  - These places are labeled with
    ## INSERT CODE HERE
    comments

SBE: What is included

- Setup and management of agents
  - I/O links created and proper callbacks
  - Tcl aliases to link Soar interpreters to environment
  - Functions for adding, deleting and restarting of agents (including necessary environment cleanup)
  - Control panel to manage agents and agent files
    - File sourcing
    - Agent naming

SBE: What is included (cont.)

- Basic window management
  - Initialization and saving of user specified postion and sizes
- File structure and file management
- Event loop
  - Simulated asynchronous behavior
  - Run until output and stop after decision support
- Commonly used things like time, time limits, score, etc.

SBE: What is included (cont.)

Management of Soar working memory

- Functions to add, remove or update WMEs easily
- Caches values so you can request that only changes are updated
- Generalized management of output commands
  - You define structure of valid output commands in Tcl array
  - Validity of output commands and parameters will be checked
  - Valid commands will add itmes to actionRequest queue
  - Proprioceptive feedback will automatically be given

SBE: What you add

- World state update code
  - Environment rules
  - Code to handle how world changes based on agent's actions
- Agent state update code
  - Update sensors
  - Translation of sensors to input link
- Any graphics/window display code

Future Work

- Translating to incrTcl
  - Fully object-oriented
  - User can then create an object that inherits from Environment and Agent objects and overload whatever functions they need
  - No need to insert code
  - Updates to blank environment will be easier to handle
- Manual

How to get

- Projects page at the University of Michigan Soar home page
- http://ai.eecs.umich.edu/soar/projects.html