



# The Future of Soar

Discussing possible paths to Soar 9 and  
beyond

# A Pre-Discussion Presentation

*Given limited resources, what aspects of Soar would benefit most from improvements?*

## Goals

- Introduce broad, orthogonal, categories for improving future versions of Soar
- Help you identify which discussion groups to participate in tomorrow
- Create new categories if necessary

# Five General Categories

- ◆ Extensibility
  - ◆ Embeddability
  - ◆ Capability
  - ◆ Usability
  - ◆ Support
- Examine categories with respect to:
- ◆ Who benefits from the changes
  - ◆ What types of changes each category covers

# Internal Changes

- ◆ Extensibility
- ◆ Embeddability
- ◆ Capability
- ◆ Usability
- ◆ Support

Modifications to the kernel

Apparent only to developers  
because the changes are all  
internal

# Extensibility

- ◆ Extensibility
- ◆ Embeddability
- ◆ Capability
- ◆ Usability
- ◆ Support
- ◆ User defined capabilities
  - Modified decision semantics
  - Modified preferences
- ◆ Interchangeable modules
- ◆ Moving to a more object-oriented design?

# Embeddability

- ◆ Extensibility
- ◆ Embeddability
- ◆ Capability
- ◆ Usability
- ◆ Support
- ◆ Enhanced API
- ◆ Better I/O interface
- ◆ Multi-thread safety
- ◆ Encapsulation
- ◆ Efficiency

# Externally Visible Changes

- ◆ Extensibility
- ◆ Embeddability
- ◆ Capability
- ◆ Usability
- ◆ Support

Adding new features to the kernel or infrastructure outside of the kernel

Benefits are apparent to all users

# Capability

- ◆ Extensibility
- ◆ Embeddability
- ◆ Capability
- ◆ Usability
- ◆ Support
- ◆ New indifferent selection
- ◆ WME activation levels
- ◆ Additional learning methods
  - Reinforcement Learning
  - Episodic Learning
- ◆ Additional memories
- ◆ *Supporting multiple kernel configurations*
- ◆ *Special purpose versions of Soar*



# Usability

- ◆ Extensibility
  - ◆ Embeddability
  - ◆ Capability
  - ◆ Usability
  - ◆ Support
- ◆ Improving/Developing tools for:
    - Creating
    - Debugging
    - Maintaining
  - ◆ New interfaces (e.g. Java)
  - ◆ High level languages for specifying behavior
  - ◆ Reusable Soar code
  - ◆ Simpler rule syntax
  - ◆ Different decision cycle semantics

# Support

- ◆ Extensibility
  - ◆ Embeddability
  - ◆ Capability
  - ◆ Usability
  - ◆ Support
- ◆ Improve:
    - Soar's web site
    - The manual
  - ◆ Develop:
    - New tutorials
    - More example agents/environments
  - ◆ Provide additional user support

# What we are asking you

- ◆ Are there any missing categories?
- ◆ For each category, what is the current state in the Soar community?
  - Visual Soar, SourceForge, gSKI, etc.
- ◆ What additional work do we need to do?
  - To make Soar easier to use (along many dimensions)
  - To make Soar more attractive to outsiders
- ◆ What is most important to work on first?