Cohen, M. A., Ritter, F. E., Haynes, S. (2004). An introduction to Herbal. In Proceedings of the XXIV Workshop. 75-77. The Soar Group, University of Michigan.

An Introduction to Herbal

Mark A. Cohen (1,2) Frank E. Ritter (2,3) Steve Haynes (3)

1 BA CS & IT Department, Lock Haven University

2 Applied Cognitive Science Lab, 3 Penn State

Project Goals:

- Make it easier to learn how to program Soar
- Promote the reuse of Soar code
- Make it easier to understand running Soar models

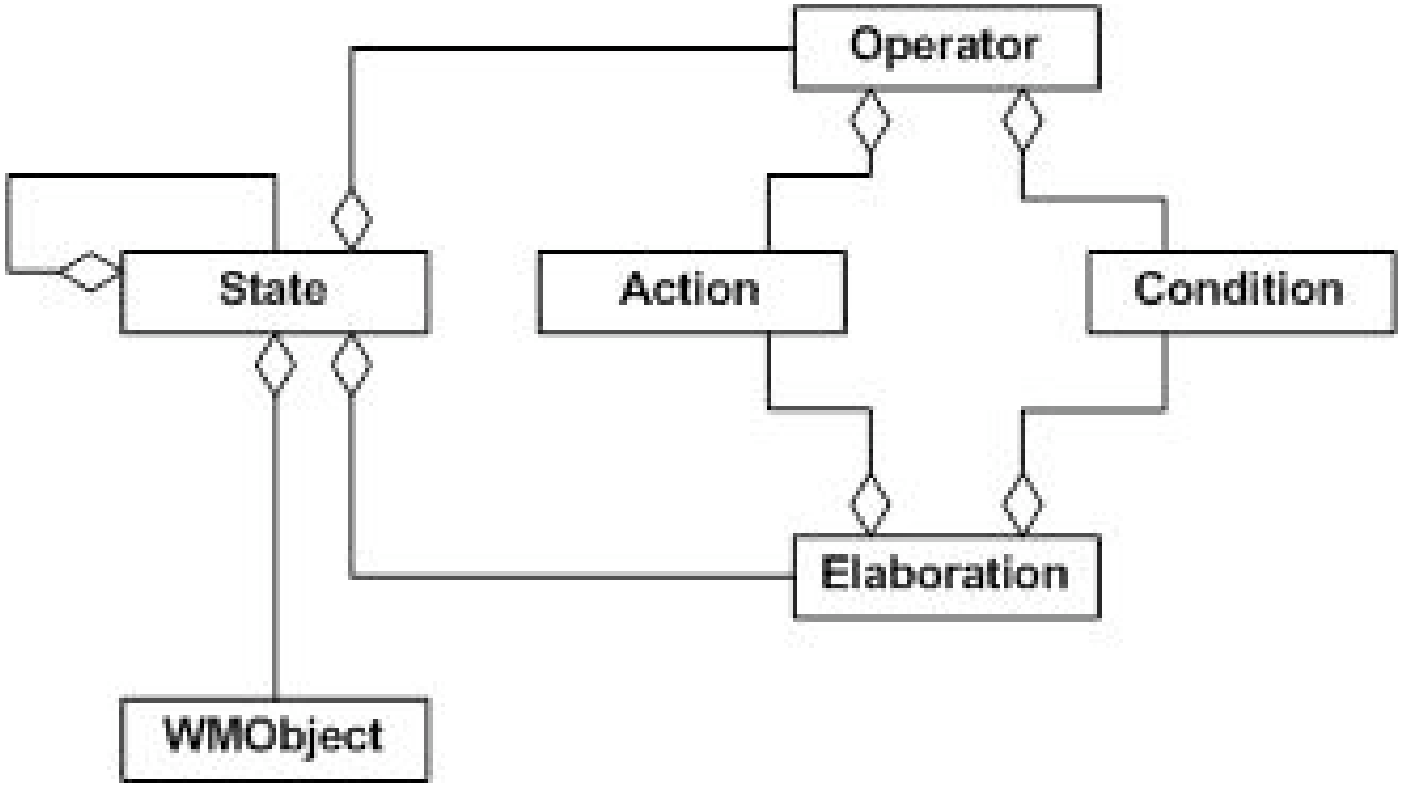
Tools that Support these Goals:

- Herbal High-Level Language
- Herbal IDE
- Herbal Viewer

Herbal High-Level Language

- The Herbal language is based on the definition of a Soar ontology
- The language is XML based (RDF) and can be compiled directly into Soar code using XSLT
- Developers can create Soar models in this language using a text editor, or with the help of the Herbal IDE

A Brief Look at the Herbal Soar Ontology



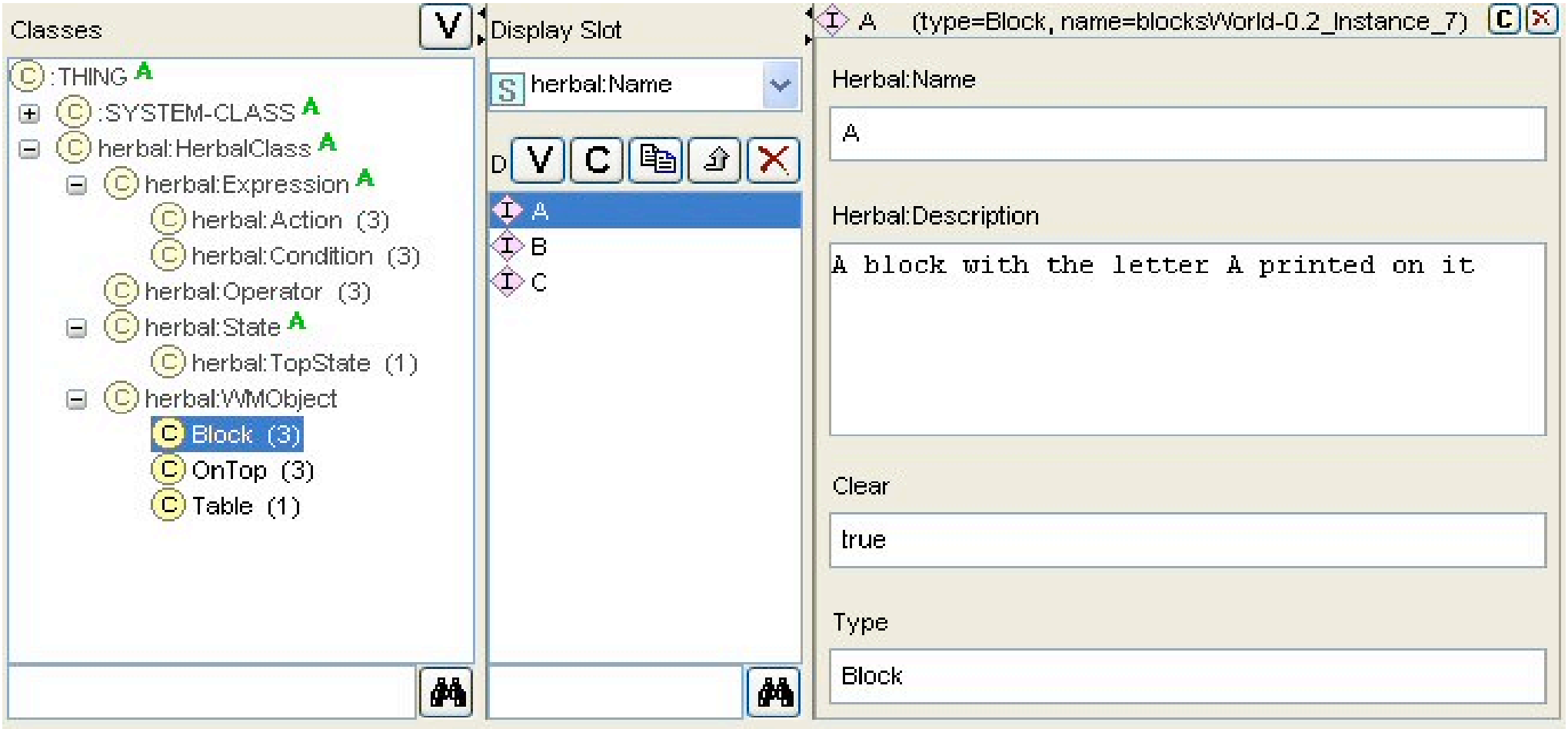
The Herbal IDE

- Used to simplify the creation of models using the Herbal High-Level Language
- Because models are written in Herbal by instantiating objects defined in an ontology, a graphical ontology editor seemed to be the best tool for the Herbal IDE -- thus Protégé was chosen for the Herbal IDE
- Protégé is a free ontology editor developed by Stanford Medical Informatics that makes it possible to:
  - Define a new ontology
  - Import an existing ontology for reuse
  - Create instances of classes in an ontology
- Protégé can be extended, as needed, via custom Java plug-ins

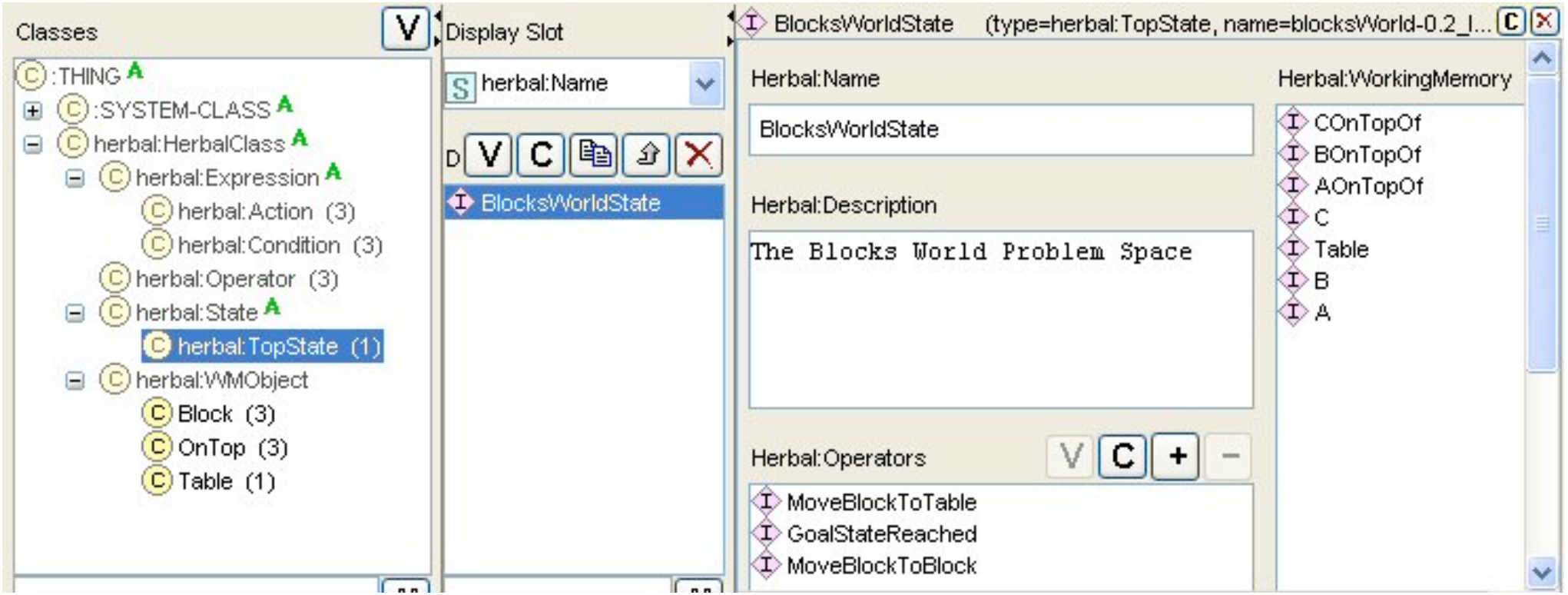
Creating A Simple Model Using the Herbal IDE

- Create a new project and import the Soar ontology
- Define and import the classes that make up your problem domain, and instantiate initial working memory
- Instantiate and import Soar objects that will define the behavior of your model
- Compile into running Soar code

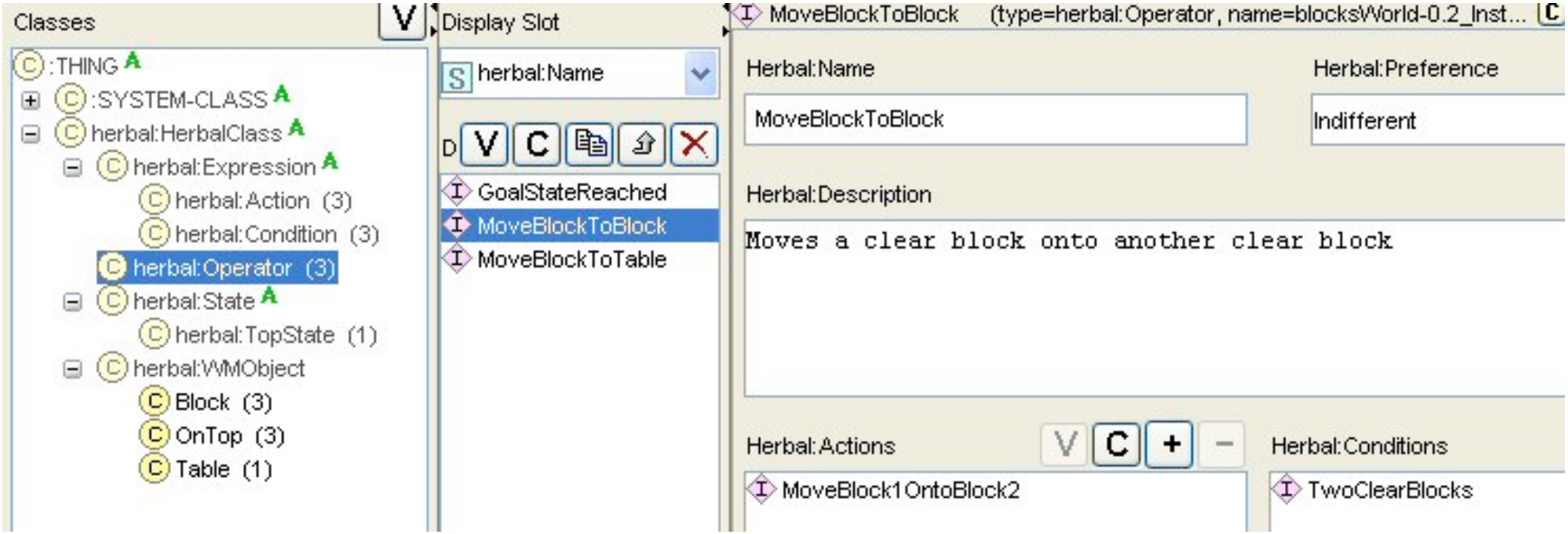
Define and Instantiate the Problem Domain



Defining Soar Objects



Defining Soar Objects (cont.)

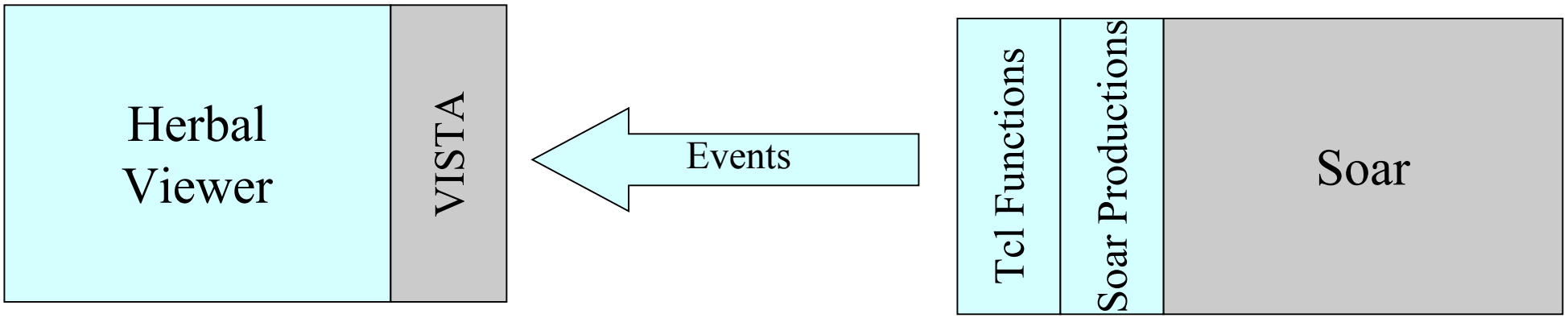


IST PENNSTATE

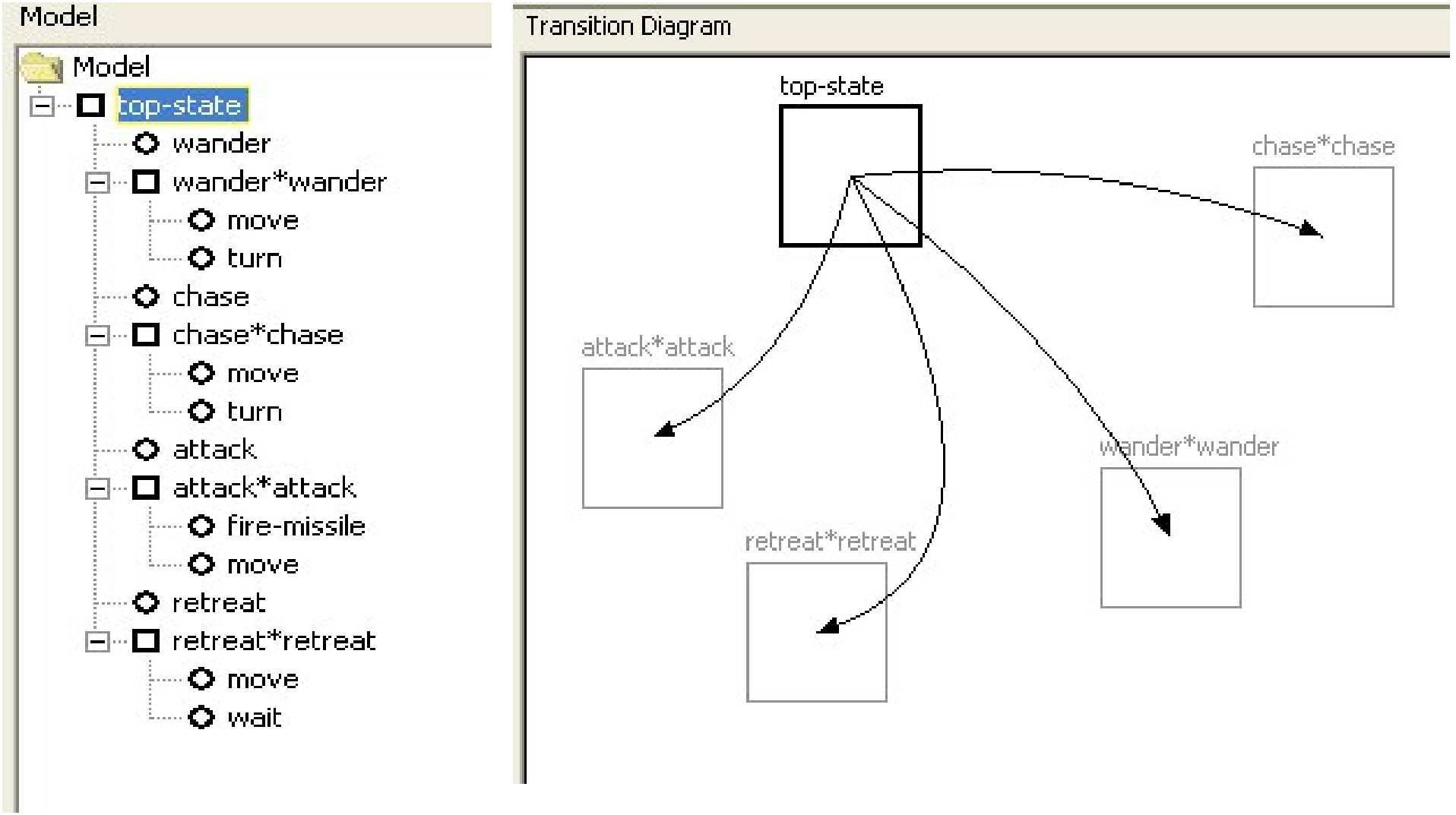
Herbal Viewer

- Generates displays of a running Soar model that will help explain the model's behavior
- Basic views work with all Soar models
- More advanced views and explanations are based on model structures from the Herbal High-Level Language

Viewer Architecture



The Tree and Graph View



IST PENNSTATE

Current Status/Challenges

- **Language**
  - We have a created a simple blocks world model
  - When I return we will start building slightly more complex models: dTank, ToH
- **IDE**
  - Protégé is serving its purpose -- we will be developing a few plug-ins for customization
- **Viewer**
  - Currently works for Soar models but does not yet use the language for creating better displays with more explanation
- **All Three Tools**
  - Will be used at PSU this fall for IST 402 Models of behavior
- **Challenges/Lumps of Coal**
  - The Soar ontology is a moving target: we are trying to keep it simple and at the same time identifying ways to inject information that can be used for explanation by the viewer
  - Personnel changes

Acknowledgements/References

- Acknowledgements
  - Isaac Council
  - Kevin Tor
  - Geoff Morgan
  - Urmila Kukreja
  - Supported by Office of Naval Res. N00014-02-1-0021

- References
  - VISTA Developer's Handbook, Soar Technology Inc. (see Glenn Taylor)
  - Protégé: protégé.stanford.edu