

TIELT/Soar

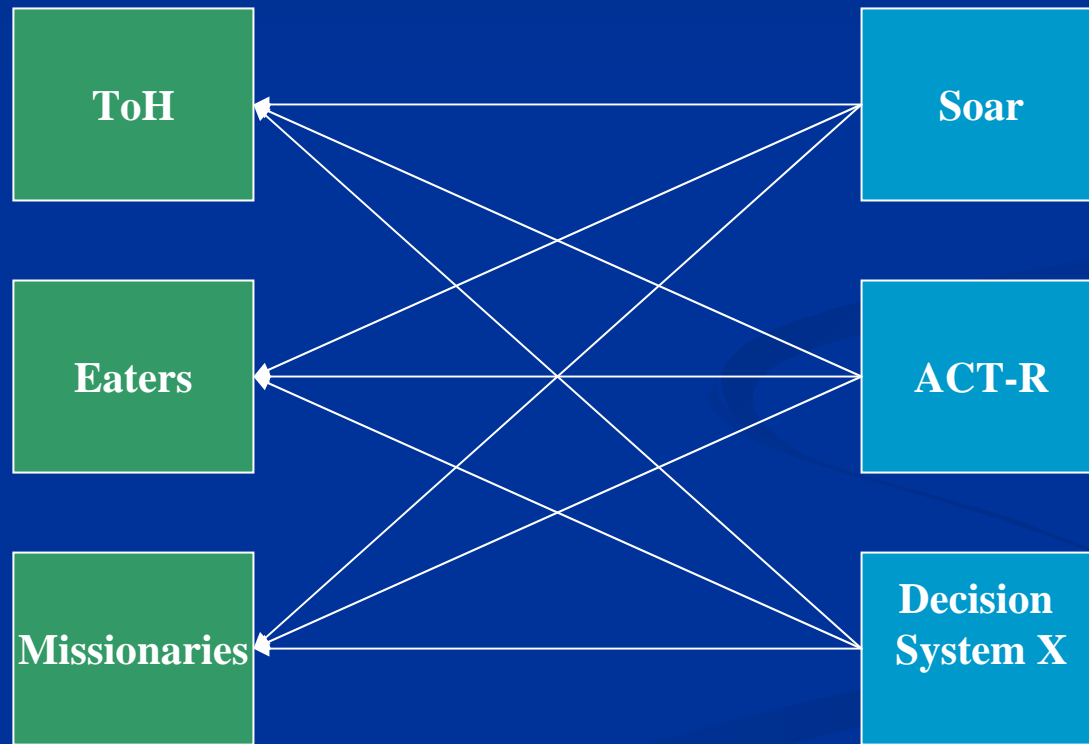
Devvan Stokes

Cory Dunham

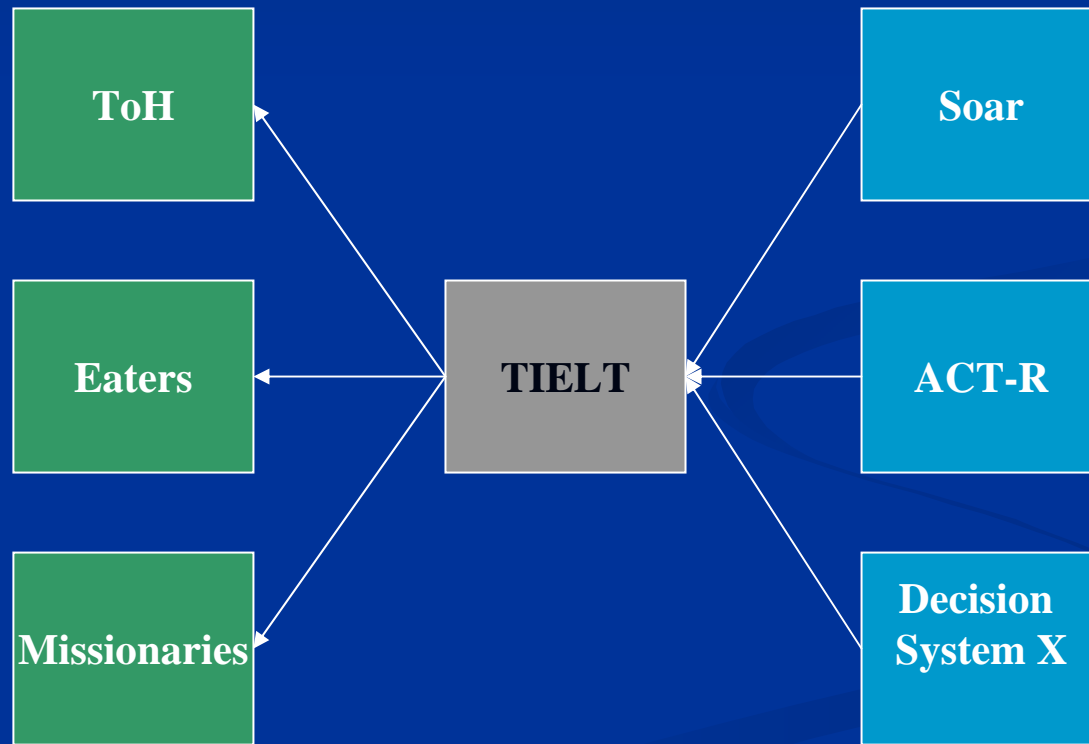
TIELT

- Testbed for Integrating and Evaluating Learning Techniques
- Alpha tool from NRL
- Goal of simplifying the integration and benchmarking of sims with decision systems

TIELT (cont.)



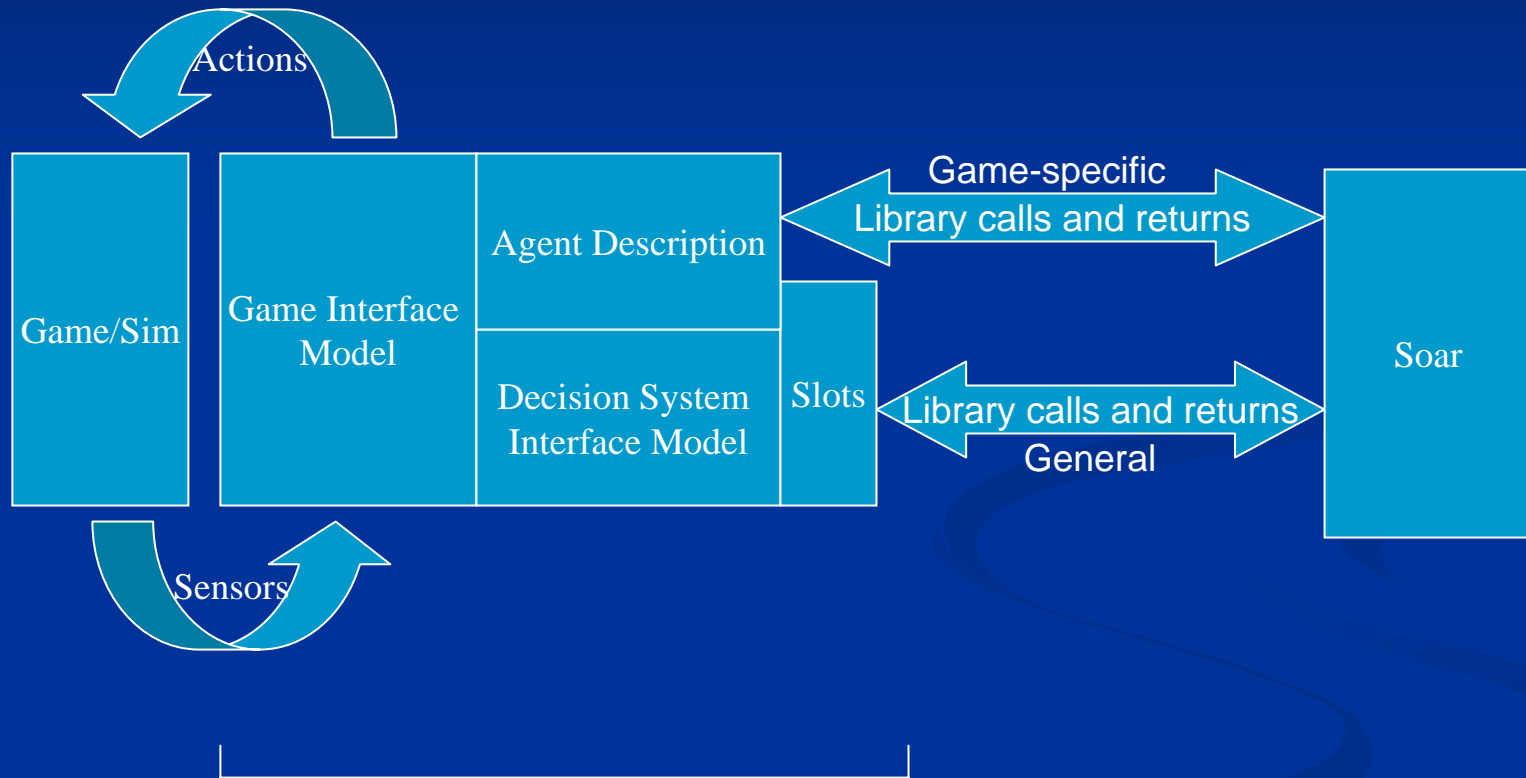
TIELT (cont.)



TIELT:Structure

- Knowledge Bases
 - Game Model
 - Game Interface Model
 - Decision System Interface Model
 - Agent Description
 - Experiment Methodology

Module Overview



TIELT Modules

Game Model

- Specifies simulation environment state
 - C++/Java-like OO spec
- Defines possible manipulations of state
- Defines phases (e.g. Turns in a chess match)
- Largely optional for Soar integrations
 - More useful for pure planning systems

Game Interface Model

- Sensors
 - Messages from sim to TIELT
- Actions
 - Messages from TIELT to sim
- Game \leftrightarrow TIELT Communication Spec
 - TCP/IP
 - UDP*
 - Library calls*
 - Input/Output streams

* (may not be implemented)

Decision System Interface Model

- Allows declaration of shared data
 - Specify “slots” to be written by decision system
 - Specify “slots” to be written by the TIELT
- Defines general functions for general Soar operations
 - Not specific to a simulation

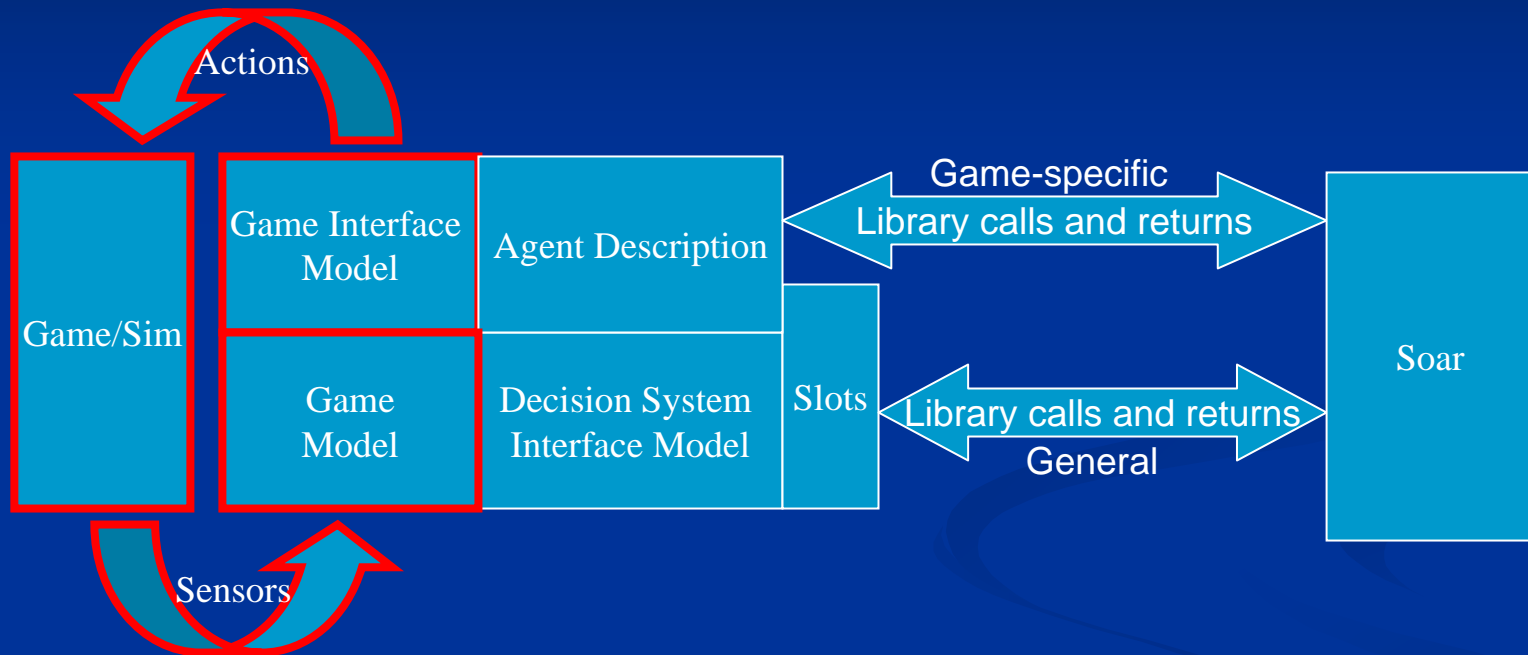
Agent Description

- Decomposes simulation into tasks and methods
- Defines functions for simulation-specific WME manipulations

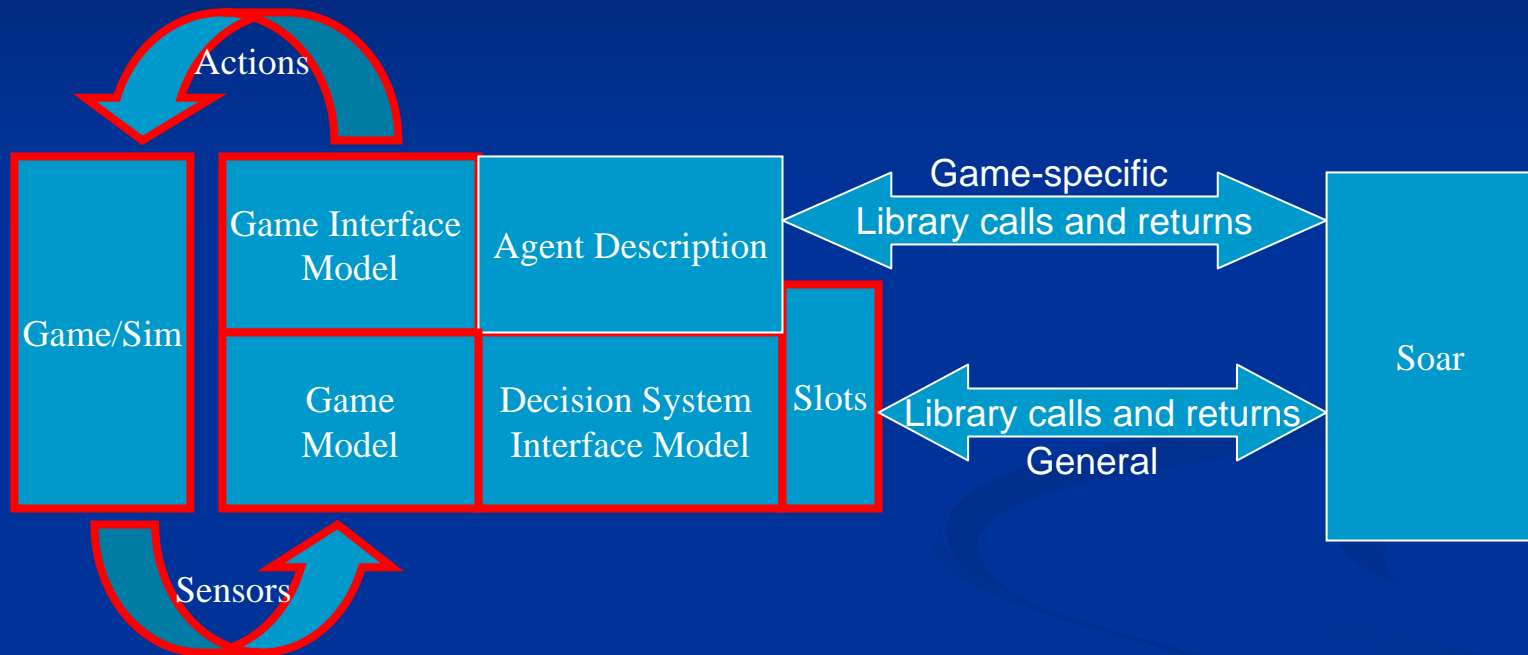
Experiment Methodology

- Defines parameters for experiment (run of sim)
 - Number of times to run
 - Which TIELT modules to use
 - Log level
- Define parameters for modules
 - Set of values for variables, etc.
- Define data capture options for stats
 - Timing info
 - External databases

Reuse : Minimum



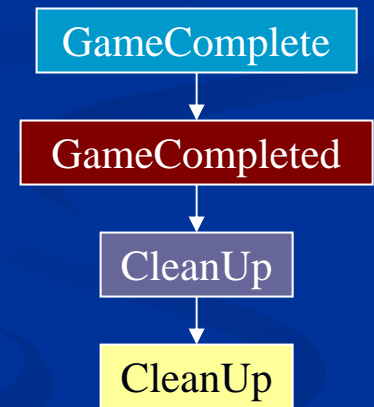
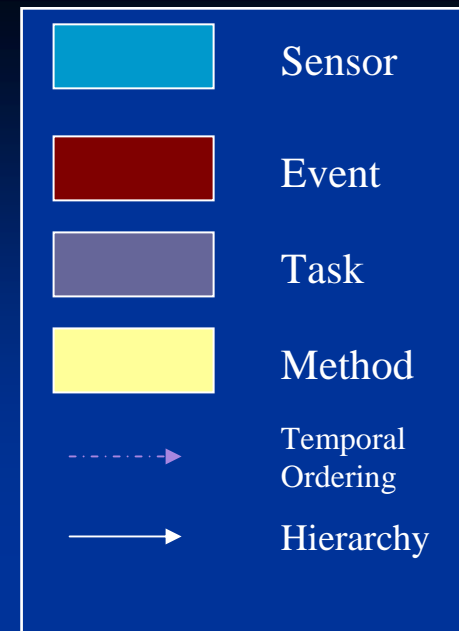
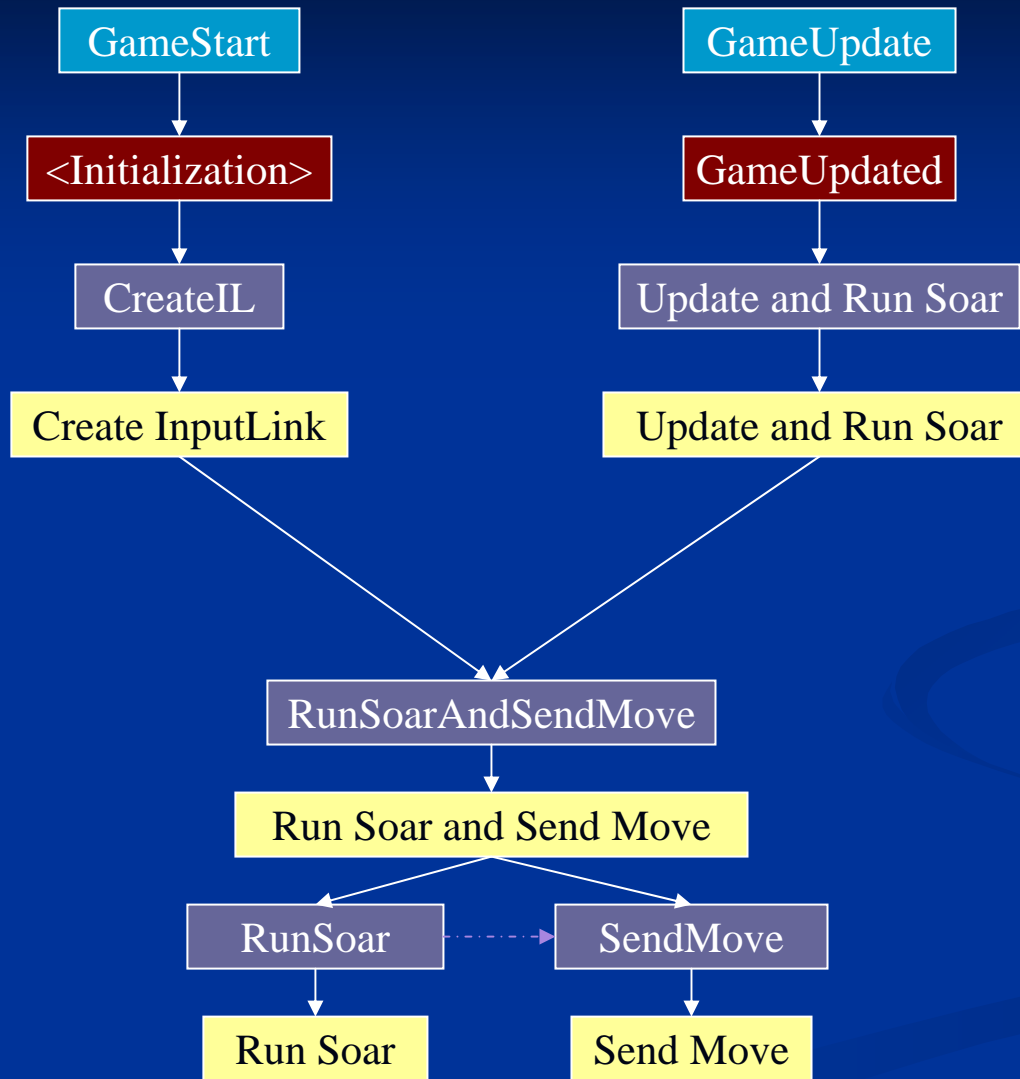
Reuse : Maximum



Towers of Hanoi Experiment

- Lines of “code” required
 - 1:1 with standalone version using SML
 - For i/o link manipulation
- Effort required
 - Using TIELT versus standalone ~ 15:1
 - Drag & Drop coding
 - May be much better now
 - Only needs to be done once

Internal Work Decomposition for ToH



General Breakdown of Work

■ Game Module

- Skip
- 1:1 for every relevant sim data structure
- Rules/Operators for data manipulation
- Hours to days

■ Game Interface Module

- 1:1 for every message that you send TIELT \leftrightarrow Sim
- Hours to days

General Breakdown of Work (cont.)

- Decision System Interface Module
 - Largely done
 - Add global data structures
 - Hours (currently)
- Agent Description
 - 1:1 for every line of interface code
 - Bulk of work, days

Future Work

- Wargus (WarcraftII modification)
- JavaTankSoar
- JavaEaters

Nuggets & Coal

■ Nuggets

- Allows a fair amount of module reuse
- Small library of completed integrations
- Polished for use with Soar

■ Coal

- Still an alpha
- Small development team