

Connecting Soar To Unreal

An Overview

By Alex Kerfoot and Devvan Stokes

UnrealScript Overview

- High level OOL
 - Modular, Java-like syntax
- Packages
 - Facilitate organized and clean development
- States
 - Group behaviors into manageable categories

Native Functions

- AddEntity(), UpdateEntity(), and RemoveEntity()
- Commit()
- ThinkSGIO()
- GetCommands()

Environmental Information

- Feelers
 - Level Geometry vs Actors
- Coordinates and Navigation
 - Nodes, doors
- Physiology
 - Health, temperature, fatigue
- Other Entities
 - Agents, items, projectiles

Tick() and Game Cycles

- Tick()
 - Called by engine every game cycle for every actor
 - Unreal has SoarBot look around and take notice of everything of interest that it can see
 - If elements of environment are different than memory, update those elements

Updating WMEs

- Changes are accumulated in SGIO's agent class
- When Commit() is called, SGIO adds all changes to input-link at once

Processing Commands

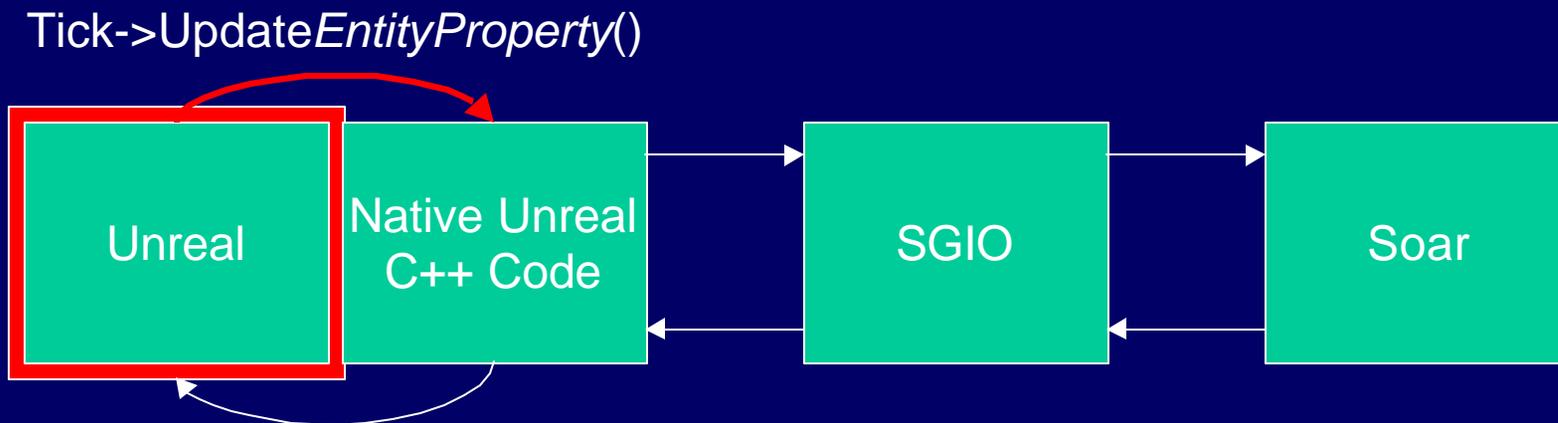
- High-level commands are issued by Soar
 - Move-to, turn-to, grab, say
- SoarInterface extracts parameters from output-link
 - Destination, angle, object, target
- SoarBot carries out commands in mid-level state code

I/O Link and Communication

- Adding attributes to the Input link requires
 - Native function declaration in UnrealScript
 - native function
UpdateAgentTemperature(int ConnId, int temp)
 - Function definition in SoarGame.dll
 - Get parameters
 - Get agent by Connection ID
 - Agent->UpdateTemperature

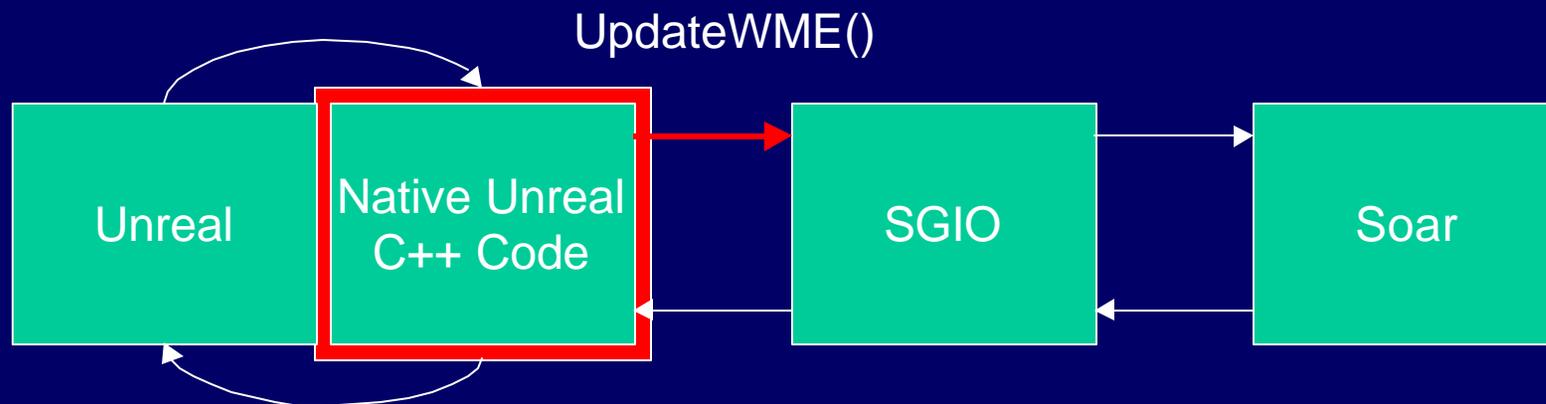
Sample Game Cycle

- Unreal bot notices entities that have changed



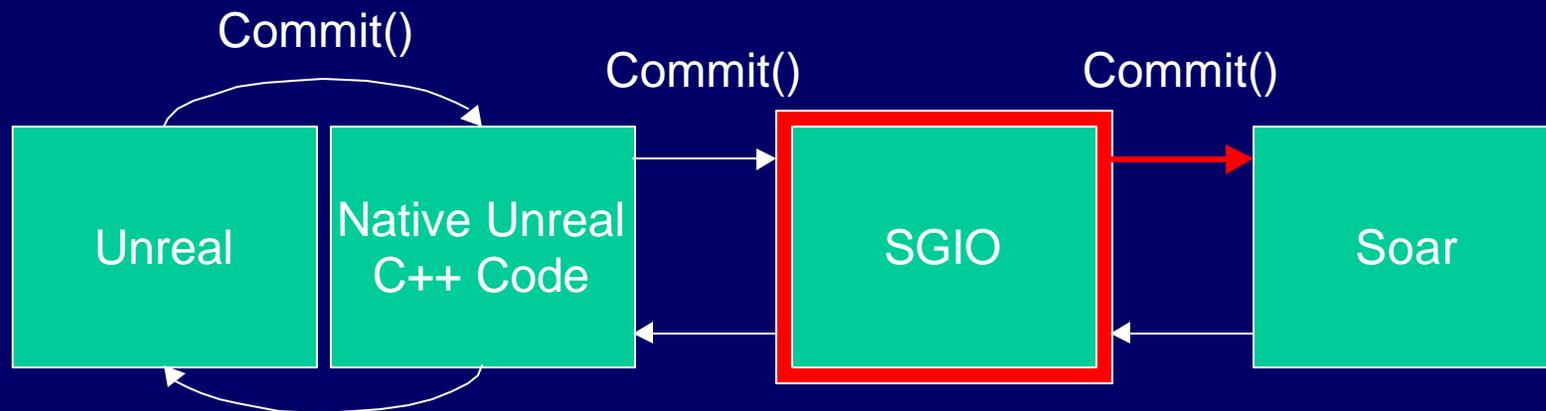
Sample Game Cycle

- Unreal bot notices entities that have changed
- **Native DLL sends updated info to SGIO**



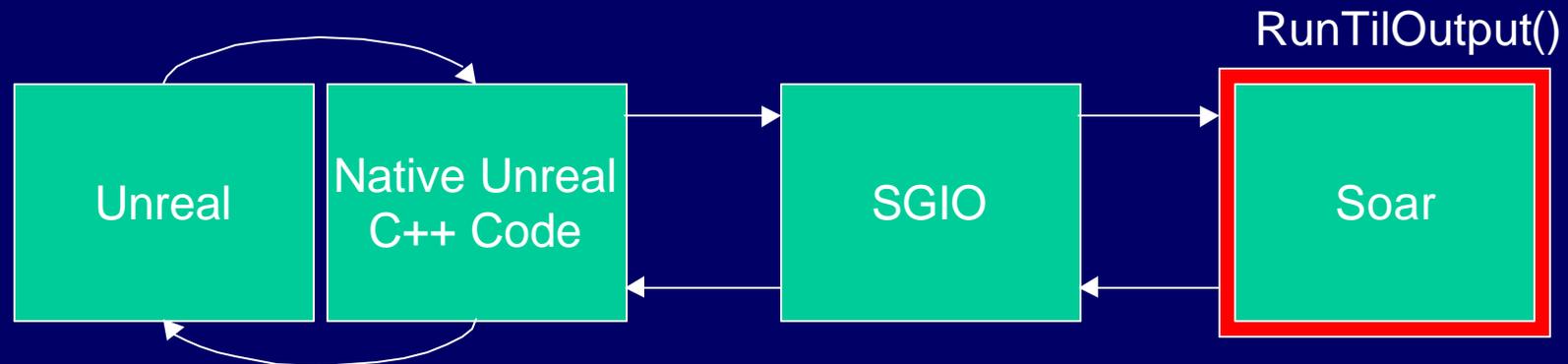
Sample Game Cycle

- Unreal bot notices entities that have changed
- Native DLL sends updated info to SGIO
- **When observation is done, Commit() all the changed WMEs to Soar**



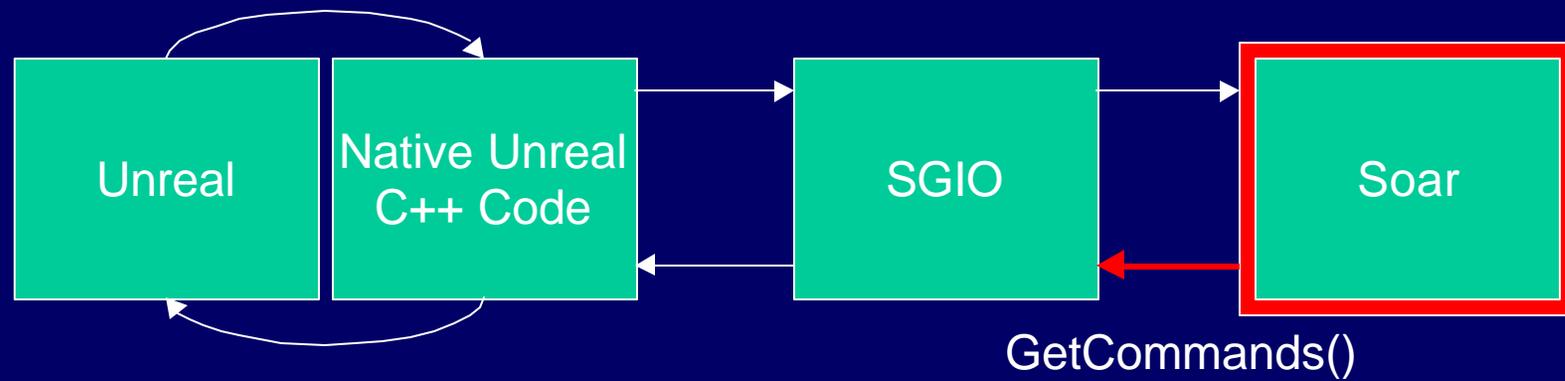
Sample Game Cycle

- Unreal bot notices entities that have changed
- Native DLL sends updated info to SGIO
- When observing is done, commit all the changed WMEs to Soar
- **Soar decides on command(s) to issue**



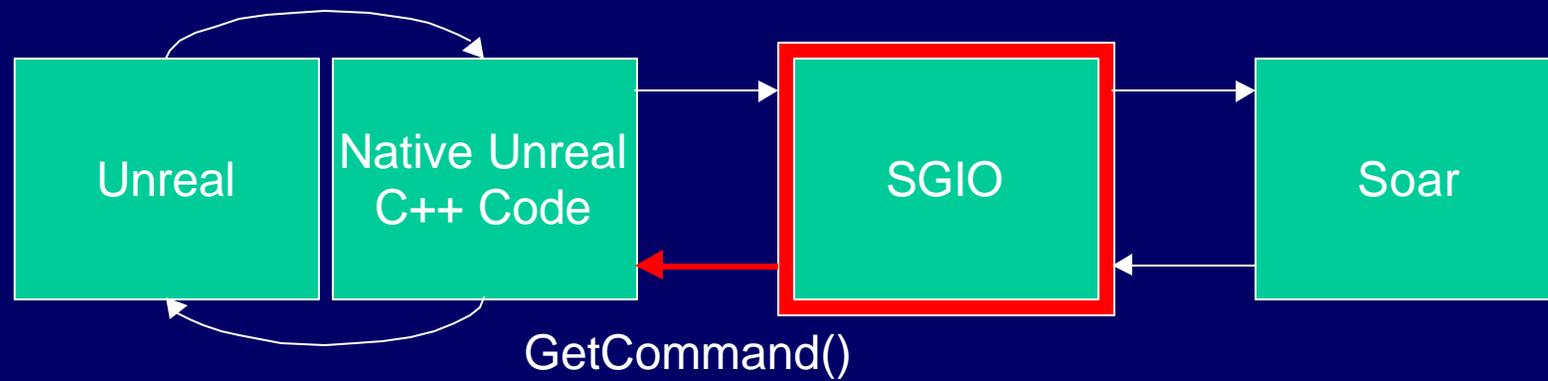
Sample Game Cycle

- Unreal bot notices entities that have changed
- Native DLL sends updated info to SGIO
- When observing is done, commit all the changed WMEs to Soar
- Soar decides on command(s) to issue
- **After decision, the actions on the output link are retrieved by SGIO**



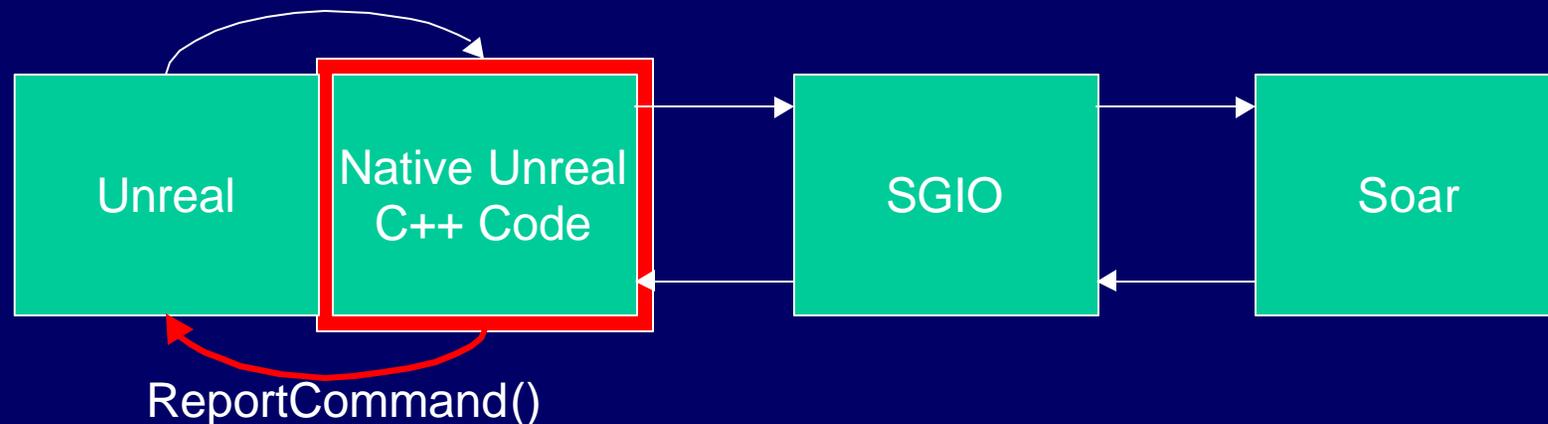
Sample Game Cycle

- Unreal bot notices entities that have changed
- Native DLL sends updated info to SGIO
- When observing is done, commit all the changed WMEs to Soar
- Soar decides on command(s) to issue
- After decision, the command(s) are retrieved by SGIO
- **Top command on the queue is passed to native code**



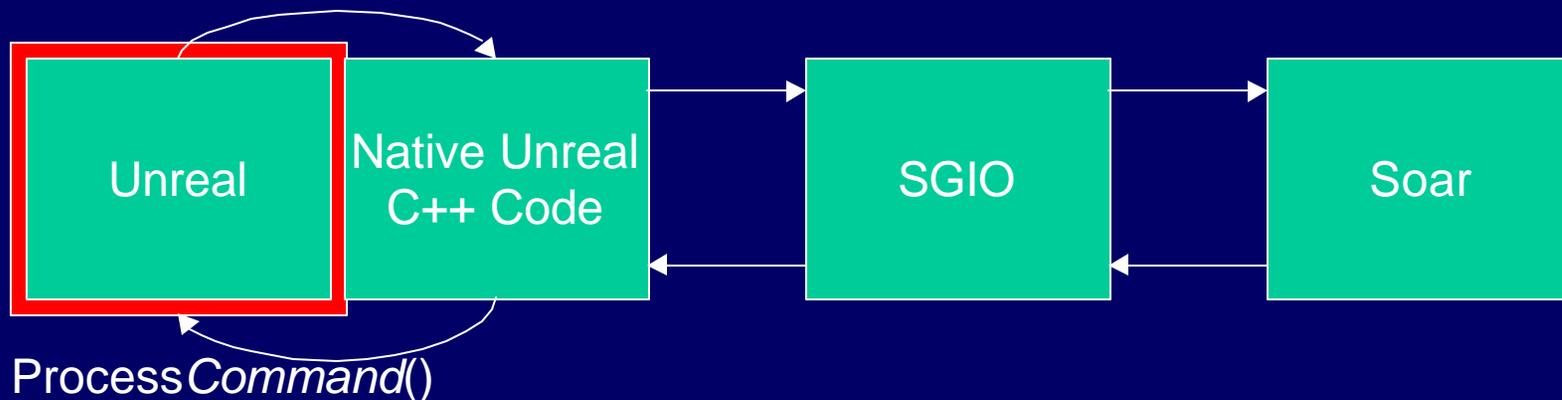
Sample Game Cycle

- Unreal bot notices entities that have changed
- Native DLL sends updated info to SGIO
- When observing is done, commit all the changed WMEs to Soar
- Soar decides on command(s) to issue
- After decision, the command(s) are retrieved by SGIO
- Top command on the queue is passed to native code
- **Native DLL reports current command to Unreal**

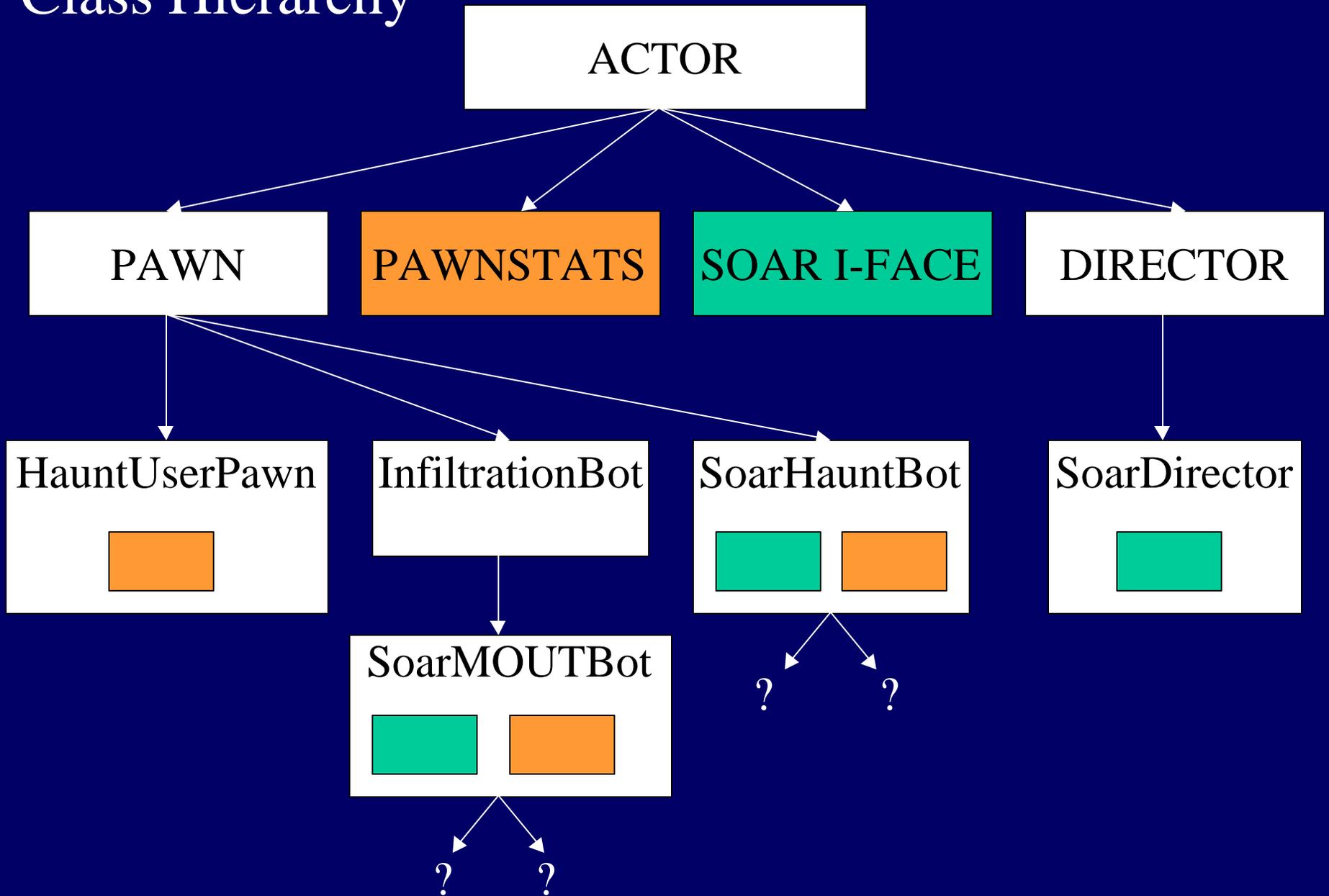


Sample Game Cycle

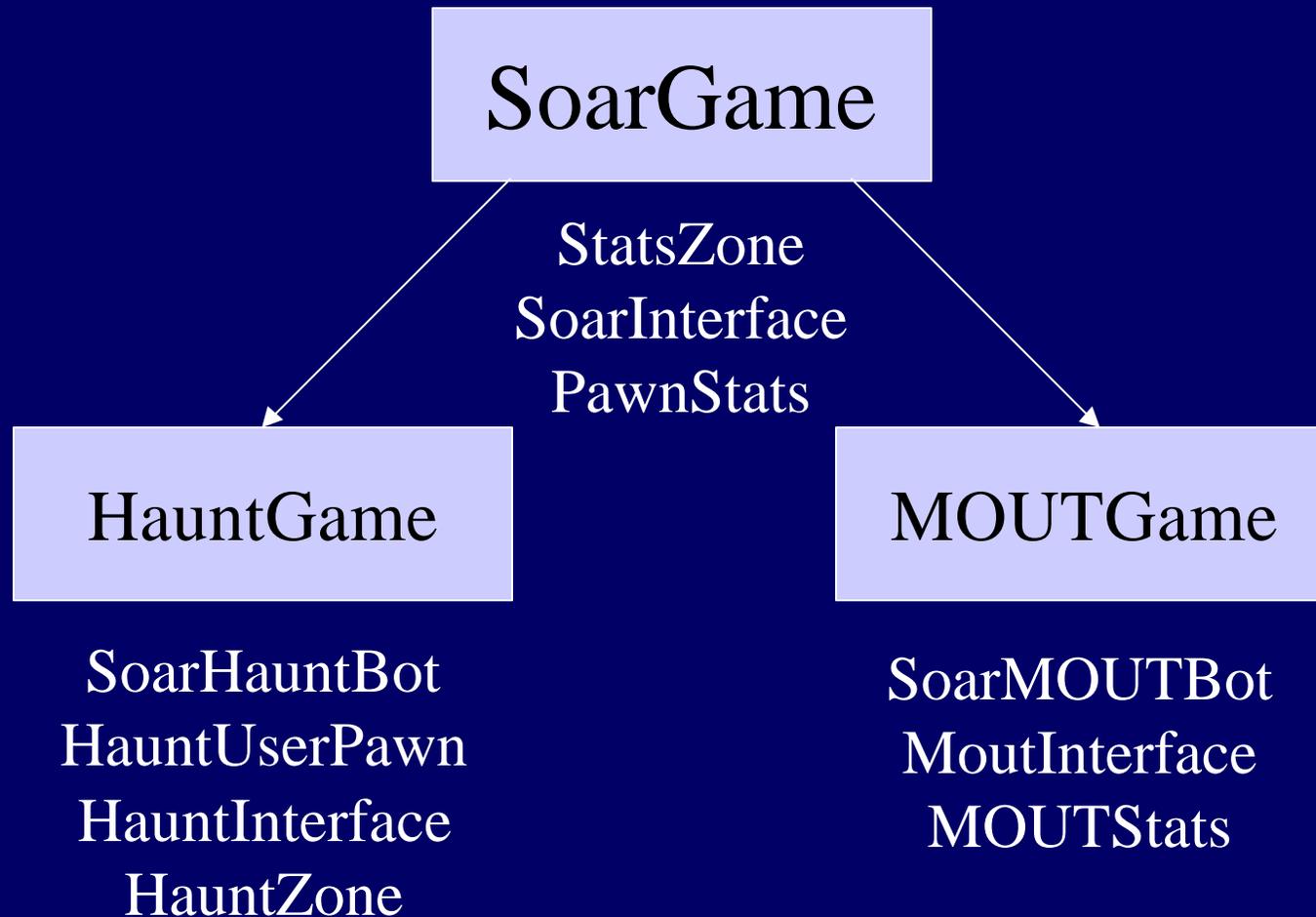
- Unreal bot notices entities that have changed
- Native DLL sends updated info to SGIO
- When observing is done, commit all the changed WMEs to Soar
- Soar decides on command(s) to issue
- After decision, the command(s) are retrieved by SGIO
- Top command on the queue is passed to native code
- Native DLL reports current command to Unreal
- **Bot performs action(s) based on command(s)**



Class Hierarchy



Package Hierarchy



Future Work

- Goal Manipulation
 - Possession

Nuggets and Coal

- Nuggets
 - Modular
 - Multiple agents w/decent f.p.s.
- Coal
 - Multiple inheritance not supported
 - Very little documentation