System features - Agent Controller

# Introduction

Being able to control any of the agents inside the engine, is what the purpose of this engine is. This Section will cover how we thought the design of an agent controller should look like. To get an overview of the classes used for the ##AgentController##, look at fig. [DOMAINModelAgentController]



[Img note: This sequence diagram shows the process of an AP taking control of agent through the ##AgentManager##, and commanding it through the ##AgentController##]

# Concept

The engine is designed to support the ability to be adapted for all APL**[FOOTNOTE: Agent Programming Language]** types, this means that the engine itself does not support all APL but instead provides a framework for quickly design of an interface between the engine and the APL.

There are two classes that one must use in order to properly design the interface:

**The ##AgentManager##** has the duty of speaking directly with the agent language it attempts to interface with, its job is to arrange it so that all agents the APL wishes to take control of is done so through it, it should do so by spawning ##AgentControllers## to all newly controlled agents.

**The ##AgentController##** is link between a single agent and the AP**[FOOTNOTE: Agent Program]**, this means its job is to take all commands directed to it and transform them into actions understood by the engine, and apply them to the agent that it controls.

The process of an AP taking control of an agent can be seen on fig. APConnectingToAndControllingAC, basically the AP calls the ##AgentManager## to locate the agent it wishes to assume control of. The agent is located through a string which is unique to it and ensures only one agent is taken. When the ##AgentManager## finds the given agent it will immediately generate a new ##AgentController##, the AP will not gain access to the agent but instead it will gain access to the ##AgentController##. Now that the AP possesses the #AgentController## it will have the ability to send the ##AgentController## commands, these commands might not be understood by the engine if the APL is foreign enough to the engine’s own language and as such it is the duty of the AgentController to convert these commands into actual actions which the engine can understand.

# Summary

The agent controller is designed very lightweight the reason for this is we do not wish to impose any restrictions that might limit an APL which we know nothing about, as such the ##AgentController## is more akin to a convention or a design pattern for how interfacing with agents should occur. It provides the skeleton of how a link might be designed but does not impose any restriction of how should link should be setup.