Implementation – Agent Controller

# Introduction

The agent controller is designed specifically to be able to accommodate all types of APL**[Footnote: Agent programming Languages]**, this meant that a lot of special care had to be taken in order for us to impose as few restrictions as possible. This section will focus on the different designs we went through and why we eventually landed on the design we have now.



[Note: This image details exactly how an AgentManager Processes incoming requests from an outside AP[Footnote: Agent Program]

# Explanation

The ##AgentManager## is designed to run separately from the engine’s model thread, which means it has the ability to take all the time needed to properly connect to an outside AP, same goes for the ##AgentController##. This means that when an ##AgentManager## generates a new ##AgentController## to be used by the AP. It also generates a new thread which the ##AgentController## is executed on. In the System features we covered how ##AgentController’s## in order to give an idea of how to use them, however we did not fully explain everything, on fig. IMPLEMENTATIONAgentManagerProccess, a sequence diagram is shown that looks familiar to the one shown in the system features. However there are a few key differences first off this sequence diagram shows the complete life cycle of an ##AgentManager##, since the ##AgentManager## is running on its own thread it does not care about blocking until work needs to be done and the only kind of work it is responsible for is ensure that ##AgentControllers## are generated for APs in need of the. Secondly it also details that ##AgentControllers## do in fact get generated by the ##AgentManager## with its own thread.

# Considerations

The ##AgentManager##, went through many design iterations in order to fully, originally the ##AgentManager## was called ##AgentServer## the reason for this is that for another language to interface with the language of the engine which is C# there must be universal way of connecting the two languages. A way in which practically no languages was prohibited from interacting, and as we thought such a way could only be achieved through a TCP connecting since the protocol for TPC connections is very old and as such is universally by almost any language. While it is true that probably almost all languages do require a TCP connecting in order for them to work with our engine. It is not true for languages that the engine understand, considering that all the .NET platform languages works together very well. The most obvious language to use would of course be F# since it is a functional language. As such if we imposed that all ##AgentManagers## are ##Agent Servers## then it would be required to setup a server just for using a language which the engine already understands.

# Summary

##AgentManager## and ##AgentController## is designed as a framework for making an interface between an APL and the engine, they are intentionally made very lightweight so that they do not prohibit the any special requirements of any given APL.